University of Bristol

Department of Computer Science

Golf Putting Trainer

Saskia Mercedes Benze

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering

September 2017 | CSMSC-17

0000048405

Declaration:

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Saskia Mercedes Benze, September 2017

Executive Summary

Current golf putting training software programs such as “SAM PuttLab” and “Quintic Ball Roll” are not accessible to the general public as they are expensive and require specialist hardware which is complicated to operate. The aim of this thesis was to create the foundation for a golf putting training application which is accessible to a wide range of users, allowing golf players to track their progress over time and identify areas for improvement.

A software program was developed which combines ball-tracking techniques, gamification elements, and data visualization methods to enable users to track their putts through video recordings, analyse the path of the golf ball towards the hole, and visualize their putting progress.

Computer vision methods including object tracking were used to detect the path of the golf ball from video footage. This component of the application was written in C++ and utilized the OpenCV library to develop the ball tracking tool. Multiple object tracking methods were compared and evaluated to identify the most suitable approach, which proved to be a combination of background subtraction to track the ball and ellipse detection to identify the location of the hole. Ground truth measurements were taken during the recording of the videos used to train and test the application so that the accuracy of the ball tracking tool could be assessed.

The data visualization component of the application, which was designed to educate users to improve their putting, was written in a combination of HTML5 and JavaScript. Elements of gamification were introduced to increase a user’s motivation to improve their putting game. Gamified educational tools have become increasingly popular within the education and sports sectors, with the technique encouraging users to develop their understanding of a specific subject or task.

The resulting application is a golf putting tool which could be used to form the basis of a mobile application, offering golf players a cheaper and more portable alternative to existing software. The key contributions of this thesis, which is 70% Type I (software development) and 30% Type II (research), are:

- A golf ball tracking tool developed using OpenCV which detects the path of the golf ball and determines whether the putt was holed successfully.
- A review and experimental evaluation of object tracking methods which identified background subtraction as the most effective method for tracking golf balls in outdoor environments.
- A method which determines the rate of deceleration of the golf ball, enabling the slope of the putting green around the hole to be identified.
- Visualizations of the ball tracking data which integrate gamification in the form of an awards system and high scores to motivate users to improve their putting accuracy over time.

Acknowledgements

I would like to thank my project supervisor, Dr. Neill Campbell, for his expert guidance, support and valuable feedback, for helping me focus my research topic and helping me prepare for the final report and poster presentation.

Contents

1 Introduction **1**
1.1 Background and Motivation 1
1.2 Aims and Objectives 2
1.3 Research Areas 3

2 Gamification in the Sports Industry **5**

3 Computer Vision **7**
3.1 Background and Theory 7
3.1.1 Colour Recognition 7
3.1.2 Brightness 8
3.1.3 Background Subtraction 9
3.1.4 Hough Transform 10
3.1.5 Ellipse Detector 11
3.1.6 Motion and Optical Flow 12
3.1.7 Perspective Transforms 13
3.2 Literature Review 14
3.2.1 Object detection using colour recognition 14
3.2.2 Combining colour recognition with a second method 14
3.2.3 Removing shadows using a colour space 17
3.2.4 Object detection using background subtraction 17
3.2.5 Combing background subtraction with other techniques . . . 18
3.2.6 Object detection using the Hough Circle Transform 19
3.2.7 Combing the Hough Circle Transform with other methods . 19
3.2.8 Combining optical flow with different techniques 22
3.2.9 Object detection and calculating velocity using optical flow . 24
3.2.10 Link to project 25

4 Project Implementation **27**
4.1 Development Decisions 27
4.2 Ball Tracking 28
4.2.1 Ground Truth 29
4.2.2 Background Subtraction 30
4.2.3 Ellipse Detection 32
4.2.4 Perspective Transform 32
4.2.5 Discussion on Ball Tracking 36
4.3 Identifying the Slope through Deceleration 36
4.3.1 Mapping Deceleration 36
4.3.2 User Practicality 40

5 Data Visualisation **41**
5.1 Gamification through graphical displays 41
5.2 Player progress 43
5.3 Further work regarding the graphs 45

6 Results and Evaluation **46**
6.1 Evaluation of the Tool's Functionality 46
6.2 Evaluation of the Tool's Performance 46
6.3 Appropriate Methods for Tracking 51
6.4 Potential Changes to Improve the Application 51
6.4.1 Multiple putts within a video 52
6.4.2 Combine two ball tracking techniques 52
6.5 Discussion on the Results 53

7 Conclusion and Future Work **54**
7.1 Contribution to the field 54
7.1.1 Contribution in comparison to previous putting tools 54
7.1.2 Contribution to putting gamification 55
7.2 Future work 55
7.2.1 Code specific calculations 55
7.2.2 Application to different areas 56
7.2.3 Further gamification 56

8 Appendices **61**
8.1 Appendix A:Video 32 - Distance from the ground truth 61
8.2 Appendix B: Ball's distance from the hole - comparison between background subtraction and the actual measurement 62
8.3 Appendix C: Deceleration of the golf ball in the first segment of the 1m downhill putt 63
8.4 Appendix D: Deceleration of the golf ball in the second (middle) segment of the 1m downhill putt 64
8.5 Appendix E: Deceleration of the golf ball in the third and final segment of the 1m downhill putt 65

1 Introduction

1.1 Background and Motivation

Over the past ten years the mobile learning tools have become increasingly popular in a variety of fields [1]. Within higher education these learning tools are considered highly successful [1] Thus, these tools have also been introduced into the sports industry [2], especially for numerous sports focussing on moving objects including football, tennis, basketball and golf. Computer vision is a popular field for analysis and covers a large range of areas, however this project solely focused on tracking and detecting moving objects in addition to image processing. Within sports there are two possibilities, tracking the ball or tracking individual players. The choice between the two depends upon whether the tool aims to identify game tactics or the position of the ball within the game.

Over the past years computer systems have gradually been introduced into sports training, however they either require expensive motion capturing systems or fail to provide adequate analysis for the end user [3]. Usually these sports training systems are based on a three-dimensional motion analysis or a two-dimensional video analysis. The two-dimensional analysis is usually affordable for the general public, however does not provide the motion analysis, whereas the three-dimensional system provides the motion analysis, but is not as affordable for the user [3]. Therefore, when developing a sports training system it is important to identify the purpose of the tool, is it supposed to only visually collect data with a two-dimensional analysis or should it actively monitor the physical motion of user with a three-dimensional system?

More and more golf amateurs understand and accept the importance of putting within the game of golf. Putting is a relatively short and low speed stroke, which aims to roll the ball into the hole [4]. Statistics revealed that for an average golfer putting accounts for 40% of all strokes during a round of golf [5], suggesting that improving putting is an effective way to lower the overall score. However, putting requires high levels of concentration and dedication, therefore practicing putting is not very popular with amateur golfers [4].

Within the golf industry a variety of ball tracking and swing analysis tools have been developed for the long shots, which analyse the golf swing in correlation to the ball flight. However, tools that focus on improving putting are limited. The most frequently used putt analysis tools are “SAM putt lab”, “Quintic Ball Roll” and “TOMI”. Both “SAM putt lab” and “Quintic Ball Roll” are tools, which require supervision by professionals, who analyse and explain the putting results to the player [6], [7]. Furthermore, both these tools are highly expensive and time consuming. On the other hand, “TOMI” is a putting tool which individuals are able to use independently to track the putter by attaching an external sensor, providing a similar but more basic analysis to “SAM putt lab” and “Quintic Ball Roll” [8]. However, all three tools do not track the ball and therefore cannot identify the distance between the ball and the hole. Additionally, these tools do not provide the player with a graphical representation of the progress made over a longer period of time. Therefore, to fill this gap in the market, this project developed the foundation for a tool, which tracks the ball, and measures the distance the ball to the hole. Additionally, if user progress is tracked over a longer period of time the user can clearly identify putting improvements. Therefore, this project creates a valuable low cost tool for golfers to use on a daily basis.

1.2 Aims and Objectives

The aim of this research project was to implement the foundation code for a golf-putting trainer, which tracks the ball's path and records the data. During the ball tracking process computer vision elements were applied to help determine whether the ball was holed or calculate the distance between the ball to the hole. Figure 1 shows three frames of a video, which were used to track the ball. Stage a and c seem to be identical, however, in Stage c the program had to identify that the putt was hole and register it as a holed putt.

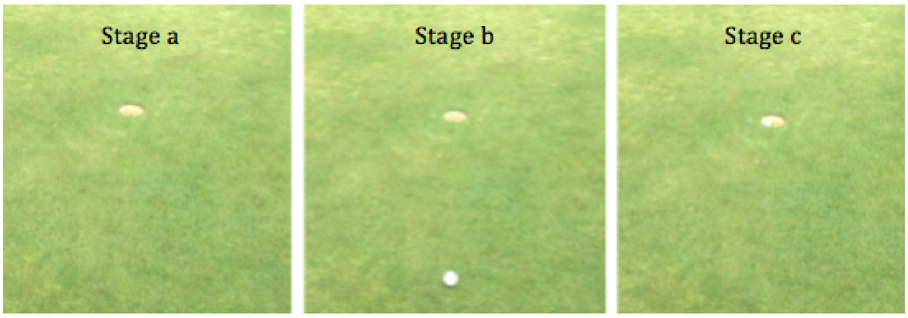


Figure 1: Stages of the Putt: a) before the Putt, b) during the Putt, c) Putt holed

Another aspect of the project was determining the ball's rate of deceleration using the data collected from the ball tracking. Once the ball's recorded coordinates were transformed into actual distances, the distance the ball travelled between frames was identified. By accurately calculating the deceleration of the ball the slope around the hole was identified as well as determining whether the putt was uphill or downhill.

The collected data was visualised using graphs, to allow the user to compare the progress made using the application. Ideally recorded progress over time would show an upward trend in balls holed and a downward trend in the distance from the ball to the hole. Additionally, a high score and star award system was introduced for the user to identify the highest number of balls holed and collect a star for each week showing improvement to the last. In order to ensure platform dependency this section was implemented using HTML5 and JavaScript.

For this research project, the OpenCV library supported the video analysis tool. Therefore, the software development was fully written in C++ allowing the tool to implement OpenCV's computer vision elements for tracking moving objects.

For this project the following objectives were defined:

- Identify and implement a suitable technique for tracking the golf ball

- Develop software which enables the application to measure the distance between the golf ball and the hole
- The rate of deceleration of the ball can be accurately computed
- The slope around the hole can be determined and visualised
- Create gamification through a graphical display of the data

The outlined objectives of this project were achieved through a large amount of programming and therefore, an in-depth understanding of C++ was necessary. Additionally, HTML5 and JavaScript were used to create a visual graphical representation of the data.

1.3 Research Areas

This project focused on two research areas: computer vision and gamification in sports learning.

Computer vision is composed of methods for acquiring, processing and analysing digital images [9]. One of its elements is image processing which uses image-to-image transformations in which both the input and output are images. Therefore, most image processing techniques can be applied to individual frames. The areas of computer vision and image processing, which will be discussed in this research project, include Hough Circle Transform (HCT), colour recognition, background subtraction, optical flow and ellipse detection. Research around these areas was necessary to fully understand the concepts of tracking moving objects and identifying which method is most appropriate in the given environment [9]. Furthermore, the research on the various methods provided insight into which method could be the most effective putting tracking tool. These methods implement areas of image recognition, which is important to analyse a pattern needed to facilitate object tracking [9], additionally, this can be used to measure various objects at once. Problems encountered in computer vision and image processing, which were faced during this project, include noisy image data, incomplete images due to shadows and moving objects in the background [9].

This research focused on the two computer vision elements: object detection and object tracking. Object detection is used within image sequences to verify the existence of the object and determine the possible location of the object within an image [10]. Whereas object tracking remembers the timing, position, size and shape changes of the object throughout an image sequence [10]. Object detection and object tracking are considered two processes, which are highly related as tracking could start with object detection. Additionally, to increase accuracy repeatedly redetecting a specific object within an image sequence often assists object tracking [10]. Furthermore, in computer vision wide varieties of applications implement these techniques such as video compression, video surveillance, vision-based control, human-computer interfaces, medical imaging, augmented reality, and robotics [10].

Gamification can be applied to improve sports learning by using game-like elements to encourage and motivate the individual to complete the task and reach predetermined goals [11]. Therefore, gamification can be seen as a goal-orientated method, which increases the users' motivation [11]. These kinds of methods have become increasingly popular within mobile fitness applications used to track their fitness activities by earning points and completing challenges [12]. Popular elements included in gamified fitness applications are, challenging the user to beat personal high scores or fostering a competitive environment amongst friends to beat each other's high scores. Even though fitness and sports seem to be very popular within gamification most academic research has been conducted within the area of gamified educational tools [12]. This project aimed to investigate how gamification can be introduced into golf, especially putting.

2 Gamification in the Sports Industry

Gamification has become an extremely popular strategy in learning through technology, influencing both the commercial and academia culture [12]. In recent years the fitness and health industry focused on gamification of mobile fitness applications (apps) [11]. The main aims of gamification include, increasing motivation and improving performance through game-like rewards and incentives [11]. Deterding, Dixon, Khaled and Nacke (2011) defined gamification as the combination of game design elements within a non-game concept. Past research within the sport sectors indicates increasing success rates of gamification [12].

Deterding, Dixon, Khaled and Nacke's study (2011) reviewed 132 gamified health and fitness apps within the sport industry. Each app incorporated an element of gamification to motivate and inspire the user. Despite the presence of gamification principles, the apps lacked professional guidance and did not meet industry standards [12]. Nevertheless, Deterding et al. (2011) results indicate that despite the lack of professional guidance the individuals improved and continued the course of the app. A similar study by Lee and Cho (2016) explored how gamification can be used as motivation in health and fitness apps. Of 142 surveys that were analysed, approximately 80% showed a successful response to improve personal fitness [13]. Additionally, the results showed that networkability, credibility, comparability, entertainment and trendiness enhanced the likeliness of the user to continue using the health and fitness apps [13].

Furthermore, gamification was tested on a variety of sports such as cycling and running. Strava is a cycling app, which records the usre's training log, presents personal times on a leaderboard and allows routes to be shared amongst friends. Barratt's (2016) study tested whether cyclists' motivation was influenced by these elements. Strava aims to change the user's attitude and behaviour towards achieving predefined goals through motivation and social influence [14]. These aims are achieved by including the following gamification elements [14]:

- Points system
- Badges awarded
- Leaderboards
- Progress bars
- Performance graphs
- Quests
- Profile development

Results from Barratt's study suggest that Strava increases user motivation to exercise and achieve predefined goals. Additionally, Strava helped users maintain daily practice habits. Fitness tracking apps such as Strava influence users to practice and publicise improvements and achievements [14]. A similar approach is used by the fitness app Nike+. Through gadgets users can track their physical activity such as running distance, calories burned, calories consumed and amount of time exercised. Furthermore, encouraging users to share their progress through social media fosters a competitive environment amongst friends to beat each other's high scores [15]. Continuous updates of the app suggest a successful approach to motivate and inspire their consumers [15] supporting the claim that gamification of a product has a significant impact on the user [16].

3 Computer Vision

3.1 Background and Theory

Object tracking and object detection are considered fundamental elements within the field of computer vision [17]. In video analysis object detection is the first step and object tracking the second [17], [18]. In order to fully understand the different techniques of object detection and tracking it is important to clarify the difference between the two concepts. Essentially, tracking algorithms are faster than detection algorithms as features from previous frames are known when tracking an object [19]. Therefore, information stored in previous frames can be reapplied to predict the object's location in the following frame [19]. Methods, which have successfully tracked objects using detection, include colour recognition, brightness levels, and background subtraction [20]. Hough transform and Optical flow are considered successful object tracking algorithms [21].

3.1.1 Colour Recognition

Colour recognition tracks objects by means of detection [10]. In computer vision colour is referred to as a linear space, which is defined by the three chromaticities red, green and blue (RGB) [17]. Depending on the illumination of specific objects their colour might change accordingly. In order to correct the colour change, varying levels of hue, saturation and brightness can be applied [10]. This reverts the object to the original colour as hue, saturation and brightness are non-linear colour space algorithms [10]. Hue is defined as the property of a colour that varies when it passes from red to green, saturation is the change which occurs when passing from red to pink and brightness varies when passing from white to black [17], [20], [22]. To detect and localise an object's colour simple tests using colour gradients can be applied. These colour gradients could be affected by noise, nevertheless certain aspects of the gradient create a unique signature that can be identified regardless of noise [23].

Another important part of colour recognition is identifying the colour boundaries of an object [23]. This is complicated by the fact that adjacent regions are rarely as sharp as the colour blends [23]. Therefore, to identify the objects boundaries, pixels, which are several pixels apart, should be compared [23]. Once the boundaries are identified a binary image is created. Meaning the relevant colours is represented as white and the rest of the image is black, reducing the image to the detected objects, meant for tracking [17]. In most cases colour recognition can be applied if there is no change in illumination. Detecting objects in an outside environment is significantly more difficult. Due to natural lighting, shadows create a major problem within object detection as they influence the perceived shape and colour of an object [21]. Hence, dealing with shadows correctly is an important aspect of image processing. Due to varying illumination levels shadows cause a change in intensity and colour [21]. A mixture of sunlight and skylight illuminates regions, which are not covered by shadows, whereas shadow regions are only lit by skylight [17]. As a result the colour temperature varies between the two regions. Applying colour segmentation to each pixel could solve this problem as it removes illumination [23]. However, this process results in information loss as the image moves into a grey-scale spectrum [23]. Therefore, shadows could be removed with colour, which is illustrated in Figure 2 [23].

(a) (b) (c)

Figure 2: Removing the shadow of a bridge using colour recognition a) original image, b) removing shadow edges, c) output image with shadow removed

As shadows hinder the detection of objects it is important to remove them. For shadow removal the following process is followed, first all the shadow edges need to be identified and then these edges need to be removed [17]. The past years a number of assumptions about this technique have been made. One assumption mentions that shadow edges have a large dynamic range; others assumed that at the shadow edges brightness level changes but not colour [23]. However, these assumptions are not accurate within every environment or situation [17]. Therefore, one needs to take the context into account in which the image will be processed.

3.1.2 Brightness

In computer vision and image processing another important method used to detect objects is image brightness, which is closely linked to colour recognition. Brightness is a key indicator of whether an object will be identified and detected [17]. Often it is simple to identify whether objects are located in bright light or shadows. However, the object is not accurately detected if its colour darkens in shadows [17]. Hence, it is important to recognise whether a colour change occurred due to reflection or shading [17].

The brightness of pixels can be analysed to detect specific objects within an image [18]. If a pixel in a reference image is selected and in the analysed image the same pixel coordinate is darker it is assumed to belong to the object [18]. The main three factors, which determine the brightness of a pixel, are reflection, camera response and illumination. The fraction of light, which is reflected from a specific surface, affects the pixels' brightness [17]. Due to a positive correlation between surface brightness and light reflection, different areas on a surface may influence the amount of light reflected [18]. Two models relevant to the theory of reflection are diffusion reflection and specular reflection. Diffusion reflection is the reflection of light, which is spread evenly across the entire object so that the incident ray is reflected at numerous angels [20]. Specular reflection, often also referred to as mirror-like reflection, means that the angle of reflection is equal to that of the incident ray [20]. Light has a unique effect on the camera, as it has the ability to produce an image with a wide dynamic range of natural light without using saturation [17]. The amount of illumination falling on a specific patch of a surface depends on the overall intensity of the light and the geometry of the object [18]. The intensity may vary as the light source could be blocked or shadowed. Geometry affects the intensity of light as the surface facing the luminaries might collect more radiation and therefore be brighter [17]. On the other hand, surfaces, which are tilted away from the light source, are darker this effect is referred to as shading [17].

Linked to colour recognition, the brightness levels are used for edge detection using the gradient of an image to compute these edges [10]. An edge identifies local properties of an image [17]. Edge detection is seen to be perpendicular to gradient detection, which points towards the direction of the image function growth [10]. Furthermore, this idea is involved in colour recognition, which was discussed in section 3.1.1.

3.1.3 Background Subtraction

Within computer vision and image processing, background subtraction is also referred to as foreground detection [18], in which a digital image's foreground is extracted for further processing. Similar to colour recognition, background subtraction focuses on the objects within an image. Background subtraction is a widely used tool for detecting moving objects within video analysis [21]. This process can be supported if the objects colour is significantly different from the background, which is being subtracted [18]. Background subtraction is sensitive to movement of the camera and changes in illumination, as these would cause a movement within the whole image [18]. An image's background can have slowly varying grey levels, which are important in the process of subtracting the background [21]. The simplest way of detecting moving objects using background subtraction is comparing two consecutive frames and identifying which pixels differ. The pixels, showing differences are identified as a moving object and appear as the foreground [18]. Additionally, a noisy image could be a complication, which might hinder the use of background subtraction [18]. The background subtraction would detect the noise as additional movement [18]. Furthermore, due to lighting issues removing the background occasionally removes elements of the object [18].

Two basic techniques for background subtraction are Difference Matting and Moving Average [20]. For Difference Matting an image with an empty background also known as a clean plate is required to be compared to the image [20]. To identify a moving object each frame is compared to the clean plate. The pixels within a frame where the colour from the clean plate, it could be classified as part of the foreground object [20]. The second method is referred to as Moving

Average. A temporal average is formed (TP), by taking the average for each pixel points in the same position within each frame [18]. As a result the background is shown with a faint version of the object. Equation 1 shows full equation for Moving Average [18]:

$$TP_xy = \frac{P1_xy + P2_xy + P3_xy + P4_xy + P5_xy \cdots + Pn_xy}{n} \tag{1}$$

Pixels in the distance of the image should be weighted at zero, and the weights increase constantly as moving to the front of the image [18]. The moving average should track the changes in the background, meaning that sudden changes in weather should result in fewer pixels holding a nonzero weight. Where as more gradual changes cause the number of pixels with nonzero weights to increase [17]. A negative aspect of this method is that even though it provides an estimate location of the object, it is only reliable when the lighting of the image is highly controlled [18]. The goal of these techniques is to estimate the location of one or more objects within a video sequence [22]. Ideally estimating the location of an object does not cause loss in tracking [22]. Often these algorithms are connected to optical flow, which is discussed in section 3.1.6.

3.1.4 Hough Transform

Hough transform (HT) is a technique, frequently used in image processing and image analysis [17], to detect lines or circles within an image [22]. Hence, if an image consists of objects with unknown shapes and sizes, HT is the most appropriate technique to use [22]. Within HT there are a number of different specializations. This paper will focus on Hough Circle Transform (HCT) a basic technique used to detect circular objects within a digital image [17]. HCT is an algorithm, which helps differentiate circles from edge-detected images and noisy environments [18]. For a two dimensional space the equation for a circle in explicit form can be described as [18]:

$$(x-a)^2 + (y-b)^2 = r^2 \tag{2}$$

Where (a,b) represent the circle centre, r represents the radius and (x,y) represent a fixed two-dimensional space [18]. As three variables are unknown a three-dimensional accumulator array is created [18]. If the radius and the circle centre are known variables then x and y can be calculated using the parametric form [18]:

$$x = a + r \cdot \cos(\theta) \tag{3}$$

$$y = b + r \cdot \sin(\theta) \tag{4}$$

One limitation of the HCT is that it requires long computation time and large memory storage as it relies on three parameters [18].

To identify the circle centre (input edge image Figure 3) the edge point method is applied. First using the identified radius the edge points are used to draw circles (parameter space Figure 3) [22]. These circles create intersection points. The point, which holds the highest number of intersections, is defined as the centre (accumulator point Figure 3) [18]. The accumulator array is used to detect the circle. Each edge point (edge pixel Figure 3) holds a set of circles within the accumulator array [22].

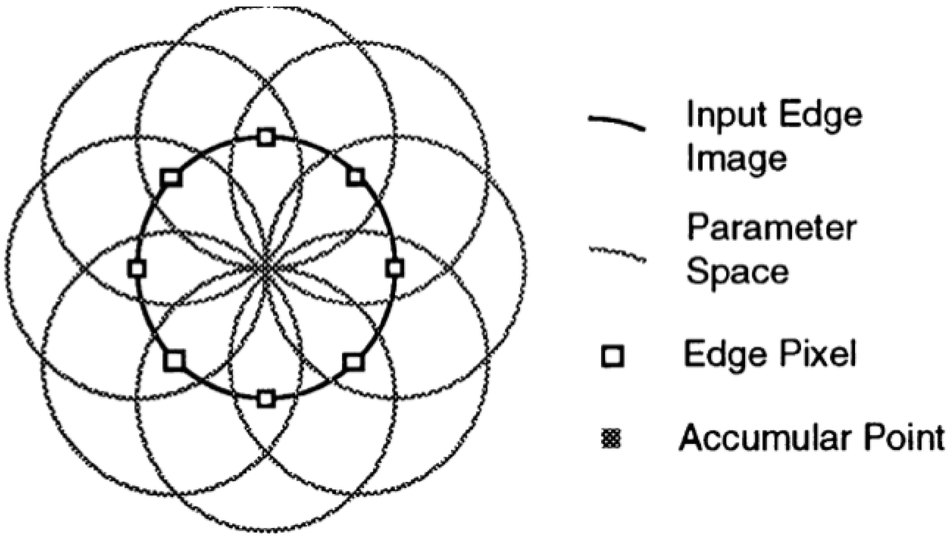


Figure 3: Identification of circle center through edge points [22]

To make this process simpler the image is converted into a binary image in which the background is black and the foreground white [18], this allows the circle to be detected using a combination of the binary image and HCT [22]. Since this process is highly dependent upon identifying the radius of the circle a program, which requires the radius to be inputted, is considered unpractical [20]. Therefore, it is important that the program identifies the radius before the circle can be detected. Overall this method is useful, however, depending on the surrounding environment of the image this task can vary in difficulty [22].

3.1.5 Ellipse Detector

Following from the HCT in the previous section a similar detection method is the ellipse detector. This method has been widely used in the filed of computer vision including gaze tracking, ball tracking, vehicle detection and traffic sign detection [24]. Often the ellipse detector is applied instead of the HCT as circles in a three dimensional space are projected as an elliptical shape in an image.

The first step of the ellipse detector is to compile a list of all edge points held by the image [24]. This list is sorted to find the points, which are antiparallel. Antiparallel points could lie on opposite ends of the ellipse. For each of these pairs the position of the centre points are identified to create a parameter space [24]. Lastly within the parameter space the positions of significant peaks are located, which allows the potential centre of the ellipse to be identified. As this method examines all possible edge point pairs it is considered to be computationally expensive. Another approach to determine the ellipse is to use different parameters. The equation of an ellipse is [24]:

$$0 = Ax^2 + 2Hxy + By^2 + 2Gx + 2Fy + C \tag{5}$$

To distinguish an ellipse from a hyperbola the additional equation can be applied [24]:

$$AB > H^2, A \neq 0 \tag{6}$$

Therefore, without any loss of data A could be rewritten as A=1. The remaining five parameters relate to the position of the ellipse, its orientation, size and shape. This approach of the ellipse detector is more popular as it is considered to have a lower computationally expensiveness [24]. Therefore, depending on the task ahead the most appropriate ellipse detector approach should be chosen.

3.1.6 Motion and Optical Flow

Motion flow is the process of determining motion vectors that define the transformation from one three-dimensional image to another [25]. It is considered a very important element of image sequences. The three-dimensional projections are known as motion flow [10]. Motion is recovered from intensity variation information of a frame sequence [25]. Generally there are three different types of motion: forward, horizontal and rotation [25]. The equivalent concept for two-dimensional images is called optical flow.

Optical flow, or optic flow, is considered the pattern of motion from objects, edges and surfaces within a visual scene [10], which is caused by the relative motion between the camera and the objects [25]. Optical flow requires the estimation of fields corresponding to the apparent motion of brightness [20]. Additionally, it is seen as an approximation of the local image motion depending on the local derivatives within a sequence of images [17]. Within the field of computer vision optical flow is used for tracking moving objects within video analysis [20].

Optical flow methods attempt to calculate the motion between two frames taken at times t and Δ t. This is done by writing each image as a function of space (x,y) and time (t) with intensity I(x, y, t) [21]. Between two frames x, y and t will move by Δ x, Δ y and Δ t [21]. If the intensity remains constant this could be expressed as [21]:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{7}$$

The important techniques associated with optical flow include motion detection, object segmentation, time to contact information, motion compensated encoding and stereo disparity measurement [25]. Furthermore, two assumptions are used when computing optical flow. Firstly, the observed brightness of any object in an image is considered to be constant over time [22]. Secondly, all points in an image plane move in a similar manner [22]. Optical flow algorithms are generally defined over grey scale images, however, could be extended into colour images by applying the appropriate non-linear colour space algorithms [20]. This is reflected by the changes in an image caused by motion during a time interval [22]. The optical flow represents the velocity field that signifies the three-dimensional motion of an object across a two-dimensional image [22]. This can be achieved by subtracting two images with the same background, which were taken at different times, to identify the difference [18]. Additionally, motion can be determined by subtracting the intensity value from the image, if there is no motion then the calculation will equal zero [18]. However, if an object in the image moves then the pixels instantly change resulting in a subtraction value greater then zero [18]. This is useful as local information within an image can be used to calculate optical flow.

Common uses of optical flow include visual effects for decreasing or increasing speed, as well as adding new textures to a moving object [20]. Optical flow is therefore a commonly used technique for detection and tracking of moving objects [20]. However, similar to most object tracking and detection techniques moving objects in the background, may be detected and falsify the collected data [20].

In the field of computer vision the Lucas-Kanade method is considered to be a differential method for optical flow, which solves basic optical flow measurement equations for all the pixels in the defined area [25]. This method also comes with a number of advantages and disadvantages. Advantages include accurate time derivatives, fast calculations and it easily compares to other optical flow methods [25]. Disadvantages are that the boundaries of moving objects are often falsely detected [25]. Furthermore, an assumption of the Lucas-Kanade method mentions that the displacements of the image between two frames are small and constant within a region of a specific point [25].

3.1.7 Perspective Transforms

To recreate the original image or frame the perspective needs to be adjusted and transformed using one of the different available methods, [26], [20] defined a perspective projection to be arbitrary linear translation, which is precisely specified through four pairs of corresponding points. Through perspective projection three-dimensional objects, which have been deformed from their original state are yielded back to the original object [21]. Estimating a perspective transform usually begins with detecting and matching image features [22]. Typically a geometric transformation includes functions such as rotation, translation, scaling and skewing. Using the wrong transformation point is considered a major source of error, because of matching errors and image noise the transformation provides inaccurate results [26]. Therefore, searching for parameters that best fit the corresponding image is an important step to ensure a robust transformation.

The transformation maps pixels (x,y) to the newly identified position (x',y') and stores the positions in vector functions [22]. A defined arbitrary matrix multiplies the vector of point coordinates, before adding a translation vector resulting in a fully completed translation[22].

3.2 Literature Review

Numerous methods and techniques have been applied in past research to optimise object detection and tracking. Often techniques are combined to exploit the strengths of each technique. The most common combination is an element of object detection with one from object tracking. However, in some cases two-object detection or two-object tracking elements were combined to enhance the result. These studies will be compared to identify which two techniques are most effective and efficient within this research environment. Generally, the papers discussed in this review deal with ball tracking in mainly tennis or football. However, in a few cases general objects were tracked or detected to test the algorithms.

3.2.1 Object detection using colour recognition

The use of colour recognition techniques proves to be very effective in ball detection. Seo, Choi, Kim and Hong (1997) used a histogram back-projection algorithm to detect the football in an image sequence. Assuming the football field has a uniform colour of green and occupies a large area of the image, the RGB histogram of each colour could be calculated [27]. Seo et al. did not exclusively use this technique for ball tracking but also to track the football player, introducing a template to identify a player. Each player was entered into a tracking list so during the analysis it would be possible to identify which team was in possession of the ball. During frame changes Seo et al. major ball-tracking limitation was occlusions between the player and the ball [27]. Additionally, due to the small ball size accurately detecting the ball in each frame proved to be problematic. To solve the problem of occlusions and small ball size, when a player was close to the ball, the player was marked, as "has the ball" [27]. Overall, Seo et al. suggested that due to occlusion issues, tracking and detecting through RGB histograms would be more suitable for cases with occlusion free ball detection.

Similarly Nummiaro, Koller-Meier and Van Gool (2003), use colour distribution in form of histograms to detect a football during a match. The weighted histogram takes both the balls colour and shape into account. The ball is tracked through the application of a particle filter by comparing the histograms of the frames using Bhattacharyya distance [28]. The Bhattacharyya distance updates a previously calculated distribution by adding the particle filter [28]. Similar to Seo et al. study, Nummiaro et al. study faced issues with occlusions. Hoverer, was able to minimize the occlusions issue by predicting strategical positions the object could appear in the following frames. Therefore, Nummiaro et al. identified that colour based tracking can effectively and successfully handle non-ridged and fast moving objects.

3.2.2 Combining colour recognition with a second method

In contrast to the previously motioned colour based detection studies, Pandya and Zaveri (2014) identified that combining colour segmentation with HT would improve the results of football tracking. Shape and colour cues were applied to each image in the sequence to detect the ball. Although Pandya and Zaveri applied a different method to the previously mentioned studies the issue of occlusions remained. In order to decrease the number of occlusions between the ball and the football pitches white lines a HT technique was applying removing all lines on the image [29].]. Figure 4 shows how Pandya and Zaveri identified the football pitches lines through HT in order to removed them.

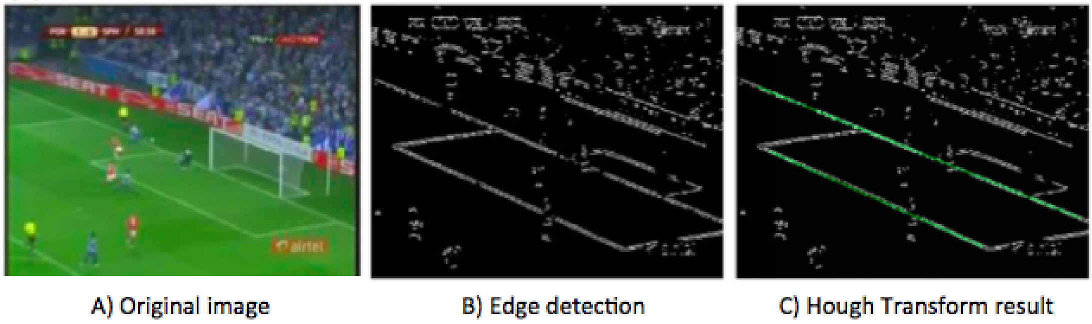


Figure 4: Removal of the football pitches lines through HT [29, Fig.10]

Furthermore, to successfully track the football frame by frame HSV was applied [29]. Pandya and Zaveri study's result suggests that tracking a moving football could be easily done by combining the HT method with colour segmentation [29].

On the other hand Pingali, Jean, and Carlbom's (1998) study came to the conclusion that ball tracking still requires improvements, as the ball was moving too fast for the tracking to successful capture each frame. Pingali et al. used a hue, saturation and value (HSV) colour space to detect and track a moving tennis ball. The HSV method was chosen due to the significant difference in colour of the yellow ball compared to the dark green background [30]. Due to the yellow colour of the ball strong hue peaks around 50 degrees were identified [30]. This segmentation was made more robust by taking the average of multiple tracking trials within different conditions. While slow moving tennis balls are successfully detected due to the high speed of the balls at professional tennis matches ball detection is inaccurate despite the use of multiple cameras. In order to identify the motion of the tennis ball a subtracting algorithm was applied [30]. Subtracting the previous frame from the current frame helped identify the motion regions [30]. The ball region was separated from other moving objects in the background using colour segmentation and the region with the highest colour peaks was defined to be the ball region [30]. Additionally, the centre of the ball was compared to the balls location of the previous frame to update the ball trajectory [30]. Although the algorithm worked for slowly moving objects as speed increased the ability to successfully detect and track the ball decreased [30], suggesting that this combination of techniques is more suitable for tracking slowly moving objects.

Similar results were found in Ohno, Miura, and Shirai's (2000) study, when the football was stationary the colour detection method worked effectively however, once the ball started moving at high speed ball detection became less accurate. To solve this detection problem Ohno et al. integrated motion information, meaning that in order to identify differences between consecutive frames motion regions were extracted [31]. Therefore, determining the ball region from a single image is seen as impossible, decreasing the value of this technique [31].

Comparably, Tong, Lu, and Liu (2004) attempted a similar approach to Pingali et al., detecting a football using colour and shapes through HSV. The two parts that the detection process consists of are field and shape extraction and, colour analysis. The football field was successfully extracted using HSV [32]. The ball was detected using the histogram, which relied on the data from the current frame to act as the reference for the following frame [32]. To allow tracking of the ball following ball detection a condensation algorithm was applied [32], which is based on sampling the ball in continuous frames.Figure 5 illustrates how Tong et al. study detected the ball through colour and shapes using different camera perspectives.

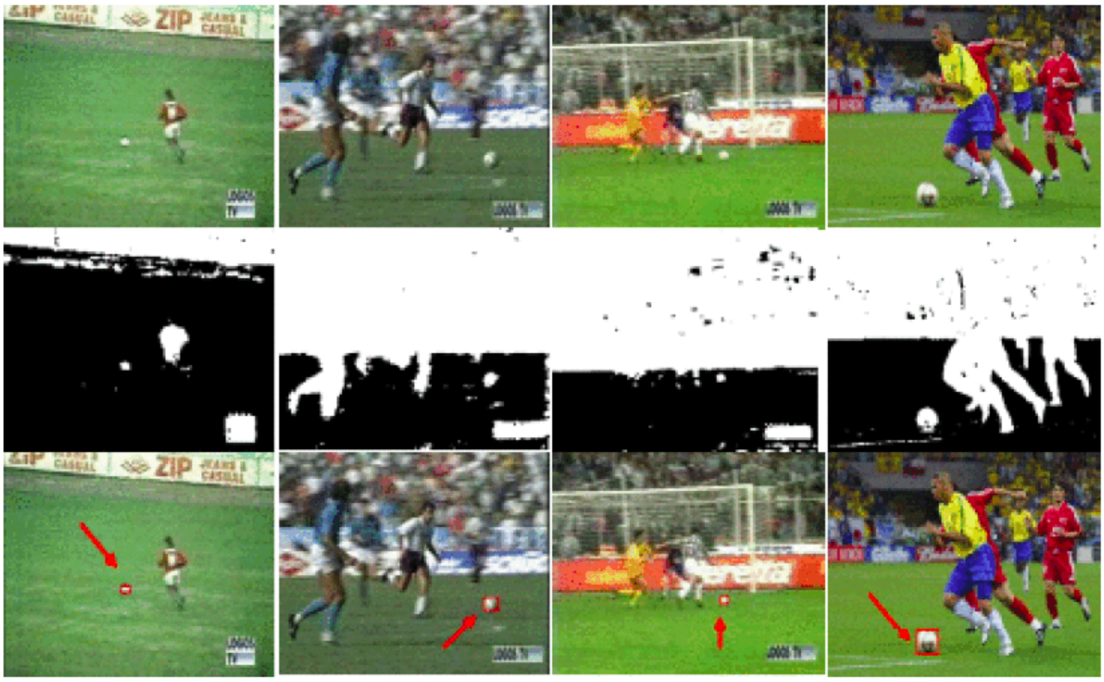


Figure 5: Shows how the football was detected in different scenarios using colour and shapes [32, Fig.3]

Additionally, Tong et al. identified that changes in movement affected the ball's size and colour suggesting that the region of the ball should be adapted. Additionally, due to the balls high speed, small size and complexity of the background ball detection proved to be difficult [32]. Even though this study combines detection and tracking methods it faces similar difficulties to Pingali et al. study due to the fast moving ball.

3.2.3 Removing shadows using a colour space

In the field of computer vision shadows are considered a major issue in ball detection [17]. A study in 2010 by Tong, Wang, and Zhang attempted to remove shadows as they often caused the ball to be inaccurately detected. It was identified that before the object could be detected the shadows needed to be removed by comparing the image to the invariant image using edges [33]. The edges missing in the invariant image represent the shadow area [33]. To create the invariant image Tong et al. changed the colour temperature of the light source. Therefore, each colour in images colour space moves in correlation with the colour temperature change [33]. Then Tong et al. compared the invariant image to the original to locate shadow edges. Setting the shadow edges to zero allowed the image to be portrayed in full colour and shadow free [33]. Once the shadows were eliminated object detection worked efficiently in an outside environment during any time of the day.

These studies suggested that using only colour to detect a moving ball is not as reliable as combining different techniques. Furthermore, studies, which implemented background subtraction, were investigated to identify how compatible this technique is with detection and tracking objects.

3.2.4 Object detection using background subtraction

Mao, Mould, and Subramanian's (2007) study investigated real-time tracking of tennis balls by applying background subtraction in each frame and additionally, integrating ball verification into the background subtraction [34]. Ball verification is an image differencing technique, where the difference between the balls position in the previous frame and the balls position in the current frame is analysed [34]. Before the ball could be detected the player movement needed to be removed from the background subtraction as shown in Figure 6.

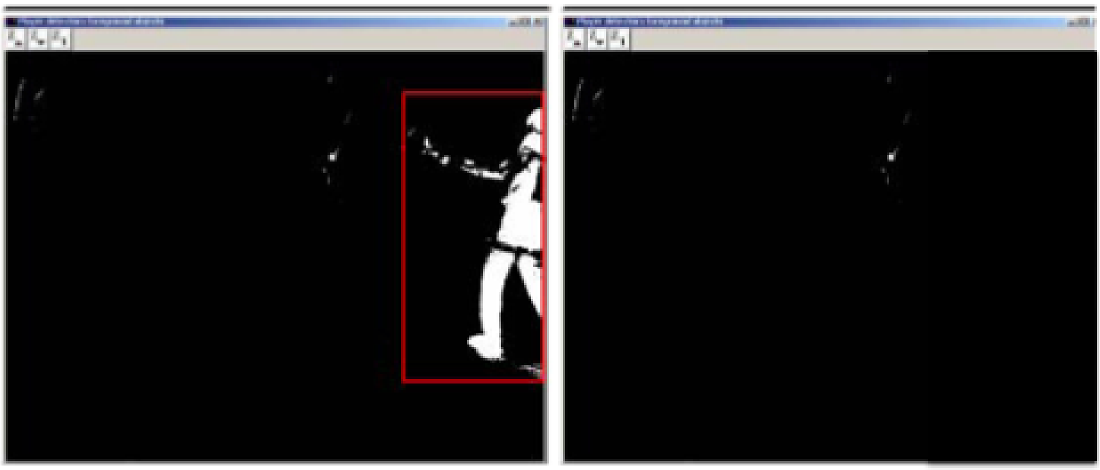


Figure 6: Removal of the tennis player [34, Fig.1]

To further improve ball detection a Gaussian distribution method was implemented for each pixel [34], allowing Mao et al. to present a framework, which was not limited to the speed and robustness of a tennis ball. Speed and robustness were the two main factors hindering the colour recognition studies discussed in section 3.2.1 and 3.2.2 to successfully detect the ball [34]. Possible improvements Mao et al mentioned included integrating multiple methods and then combining the strengths of each method [34]. Overall Mao et al. results achieved 90% accuracy in fast tennis ball tracking.

A similar approach was taken by Archana and Geetha's (2015) study, which used background subtraction to detect and track a fast moving tennis ball. Additionally, Archana and Geetha identified that noise issues within the image will interfere with the background subtraction causing inaccurate data collection. Additionally, due to the tennis balls small size the background subtraction method often identified the ball as noise. Therefore, a modified background subtraction approach was applied to compensate for the limitations of the low quality image frames and small ball size [35]. Ball detection was achieved through frame differencing between the previous frame and current frame [35]. The approach of frame differencing was chosen as tennis balls are fast moving and therefore hold an entirely new set of pixels in each frame [35]. To reduce noise levels the image was smoothened by applying a blur filter called "median filter". After these steps the background subtraction method was applied to detect the edges of the objects located in the foreground. The overall success rate of this approach was the same as Mao et al. at 90% [35], both significantly higher than the previous studies using colour recognition techniques for object detection.

3.2.5 Combing background subtraction with other techniques

Comparably to Mao et al. and, Archana and Geetha, in Gomez, López, Link and Eskofier's (2014) study, background subtraction was applied to facilitate ball detection. However, for ball tracking Gomez et al. combined the background subtraction method with the HCT method [36]. Firstly, background subtraction was combined with a foreground extraction algorithm to detect the location of the ball [36]. Furthermore, the ball detection process was able to identify illumination changes from one frame to another and act accordingly to keep the ball's image and shape constant for the HCT. The overall result of this approach only showed a 54% successes rate, which is a lot lower than the 90% success rate of the previously discussed studies by Mao et al. and, Archana and Geetha who only applied the background subtraction method [36]. Gomez et al. suggested the reason for the lower score was caused by the volleyballs high velocity and spin which changed the projected ball into an elliptic shape, which caused the HCT to experience greater difficulty in tracking the now elliptic volleyball at a standard 25-frame/second video [36].

3.2.6 Object detection using the Hough Circle Transform

Similarly to Gomez et al. and, Pandya and Zaveri who combined the HCT method with another object tracking techniques, Djekoune, Messaoudi, and Amara (2017) introduced a modified version of the HCT method for iris detection. This modification used the basic HCT and adapted it to localise the pupil boundary. The results showed that the iris was easily detected however, this was expected as the iris is not a fast moving object nor does the background create any limiting movement as the images consisted mostly of the eye as show in in Figure 7 [37].

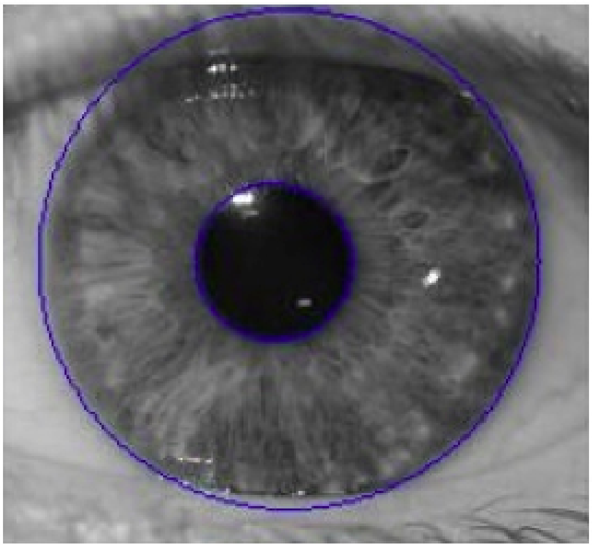


Figure 7: Representation of the image used for iris detection [37, Fig.17]

Additionally, to implement this technique the study used MATLAB and C++, which are considered to be highly compatible with the HCT [37]. Even though the objected was easily detected using the HCT the computational cost was high and the time performance was low.

3.2.7 Combing the Hough Circle Transform with other methods

Similarly to Pandya and Zaveri's study, Ioannou, Huda, and Laine (1999) used a two-dimensional HT and a one-dimensional histogram approach to detect footballs within an image. The HT was used to detect the balls centre by identifying that a line, which bisects any given point of the circle perpendicularly, must pass through the centre as shown in Figure 8 [38], and therefore, detecting the circle even if the object is not fully present.

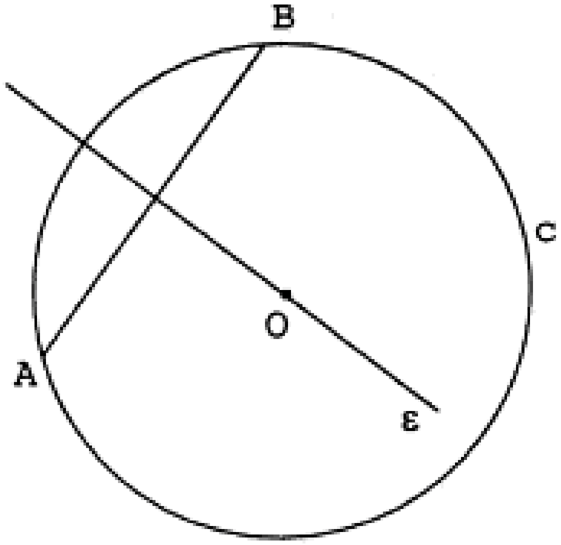


Figure 8: Identifying the circle centre through perpendicular bisection [38, Fig.1]

The second part of the algorithm involves one-dimensional histogramming. Unlike in Pandya and Zaveri study, Ioannou et al. use the histogram to detect the circle centre in addition to calculating the radius. Overall showing that the combination of these algorithms can be applied to images to detect footballs of different sizes in various environments [38]. However, in this case the football was only detected and not tracked, therefore eliminating the difficulties previous papers faced regarding balls speed and shape changes. Likewise, Rizon, Haniza, Puteh, Yeon, Shakaff, Rahman, Sugisaka, Sazali, Rozailan and Karthigayan's (2005), study also only focused on object detection, aiming to identify circles within an image by applying the HCT combined with a image separability filter [39]. First the image processing technique separability filter is applied to the image, which is a template-based method decreasing the images computing time [39]. As the radius's of the circles were known, an output accumulator space was applied, using edge points to create circles with the known radius [39]. The pixel with the highest number of intersections from the accumulator space was assumed to be the circle centre. Unlike Mao et al. who found the HCT to be unsuccessful, Rizon et al. detected the objects at a success rate of 96%. To allow a comprehensive comparison of this success rate to object tracking further tests would have to be conducted and evaluated.

In D'Orazio, Ancona, Cicirelli and Nitti's (2002) study the HCT was combined with background subtraction to track a football, aiming to find circular patterns given the balls radius in an image. Similar to Rizon et al. study each edge point aided the output accumulator space to identify the centre of the circle [40]. However, a few difficulties were identified including occlusions, shadows, and real time processing. Due to illumination difficulties, in each frame the white ball appears only as a semicircle [40]. Therefore, the image was invariant meaning the overall colour temperature of each frame varied [40].Figure 9 shows the ball before and after the image processing.

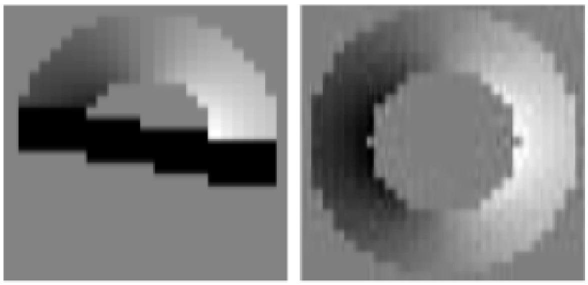
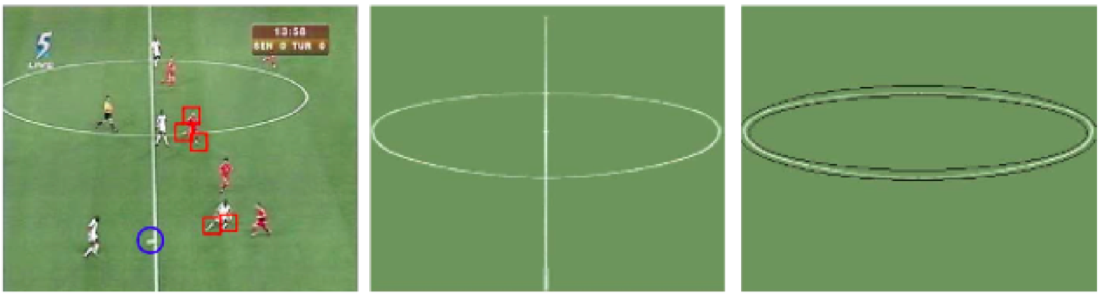


Figure 9: Ball detection before and after image processing [40, Fig.5]

Background subtraction was applied to display the moving ball in the foreground and remove any noise, which interfered with the tracking [40]. The ball tracking results were successful, however the unsolved issues about successfully removing shadows and decreasing occlusions remained [40]. Therefore, a follow-up study was conducted by D'Orazio, Guaragnella, Leo, and Distante (2004) building upon the findings from D'Orazio, Ancona, Cicirelli and Nitti's (2002) study using background subtraction for ball tracking with the aim of identifying further techniques which would limit the margin of error. A major problem was found regarding the computation, as due to the high memory storage the HCT requires this method would be highly expensive [41]. To decrease the computation problem a circle detection operator was applied across all the pixels, producing a maximum value [41]. The circle detection operator used the radius to identify the circle using a radius range [Rmin; Rmax] [41]. After removing the limitations identified in the previous study, the success rate increased from 75% to 92% accuracy [41]. Therefore, to be able to combine the HCT with background subtraction a circle detection operator needs to be applied decreasing computational problems and increasing overall success rate.

A further study by Atherton and Kerbyson (199) investigated the two image processing ideas: reducing noise within an image through a filter and applying HCT to detect and track a football. Similar to the studies of Ioannou et al., Rizon et al., D'Orazio et al. and D'Orazio et al. an edge detection method was applied to use the output accumulator space to identify the centre of the circle [42]. As edge detection was successfully applied throughout Atherton and Kerbyson's study the edge method seems to be reliable for identifying a circle's centre. Even though a combination of noise reduction and HCT was applied, Atherton and Kerbyson's only a few footballs were considered as successfully tracked. Facing a similar problem as previous papers increasing ball speed resulted in lower accuracy levels in ball detection accuracy. Atherton and Kerbyson suggested that in order for the HCT to increase the success rate the HCT would need to be modified to perform transformations including scaling, rotating and transforming the image.

In Yu, Wai Leon, Xu, and Tianstudy (2006) study HCT was used to detect the football and an ellipse detector was applied to identify the centre circle on the football pitch (see Figure 10). Yu et al. applied the ellipse detector because finding the centre circle required a fast algorithm in order to detect the centre of the football pitch without requiring a high computational time [43]. The football pitches centre circle needed to be detected accurately so that the HCT would not confuse it with the ball position and thus falsifying the study's data [43].



a) Original image b) After removal of objects C) Ellipse detection result

Figure 10: Ellipse detector identifying the centre circle on the football pitch [43, Fig.5]

Mai, Hung, Zhong and Sze (2008) study compared the ellipse detector with the HCT to identify which technique is best suited to detect different circular objects. As images take a three-dimensional environment and portray it on a two-dimensional image the objects and surroundings become deformed in size, shape and position [44]. To test that their ellipse detector worked successfully Mai et al. used semicircles next to the ellipse objects. Different objects such as traffic signs, tennis balls and car wheels were tested. The results suggested that in contrast to the HCT the ellipse detector did not require as much storage due to less parameter space required [44]. Less storage results in a decrease in computational time meaning the ellipse detector is a less expensive method to apply [44].

3.2.8 Combining optical flow with different techniques

Optical flow is an image processing method discussed in section 3.1.6. Wang, Ablavsky, Shitrit and Fua (2014) investigated the effectiveness of combining colour recognition and HCT to detect round objects. The main technique implemented in this this study implemented was optical flow. Optical flow tracked the football during the game to identify any differences between the ball being in the air to when the ball it on the ground [45]. Wang et al. identified that frame-to-frame tracking was very unreliable due to occlusions, which remain an unsolved problem throughout the study. Therefore, Wang et al. concluded that this type of tracking would be more beneficial for individual or one-on-one sport games such as tennis, golf or shot-put because the likelihood of the ball being hidden by other players on the field is unlikely, which would reduce the risk of occlusions.

Comparatively to Wang et al. study, Aires, Santana and Medeiros's (2008) study combines colour information and the Lucas-Kanade method to improve the optical flow estimation. Aires et al. study assumed that small regions within an image, which correspond to the same object, have a similar movement as illustrated in Figure 11.

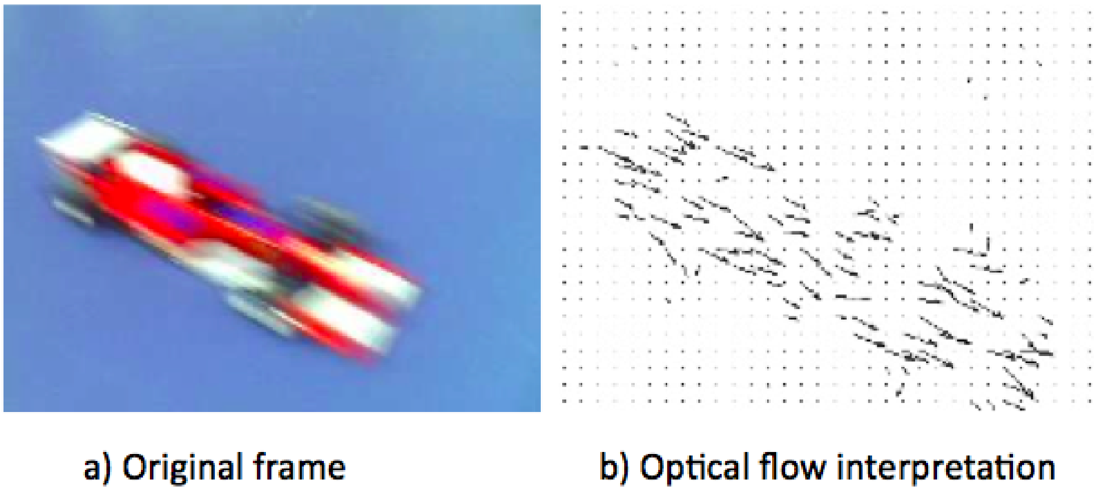


Figure 11: Movement within a toy car region represented through optical flow [46, Fig.5]

Assumptions about the movements in the image were necessary, as the optical flow method could not determined the exact movements after implementing a greyscale to one image sequence [46]. Additionally, Aires et al. added 30% of valid estimations when using colour information in comparison to no colour information [46]. Even though Aires et al. study showed promising results detecting the object using colour recognition, where as the optical flow section of the study used for object tracking did not show successful results [46].

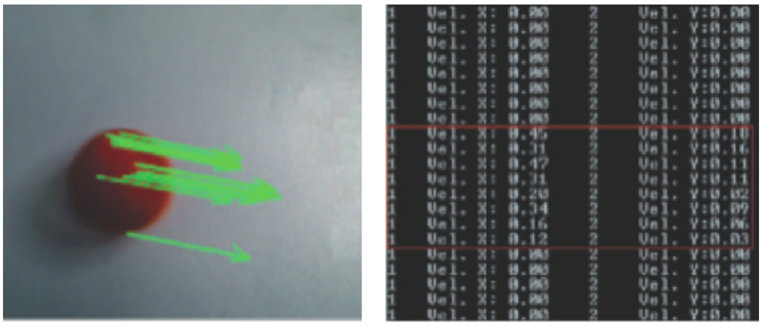
Chauhan and Krishan's (2013) study followed a similar approach to Luo et al. study by applying an optical flow method. However, Chauhan and Krishan combined the optical flow method with a Gaussian Mixture Model (GMM), which is regularly used in complex environments as it is seen to be a soft clustering method, which calculates the probability of each data point before it is being assigned. Optical flow is used for quick calculations with simple backgrounds as well as to calculate motion between two sequential frames by identifying certain points within each frame [47]. Even though the GMM was not considered to be a complete object-tracking tool, in combination with the optical flow method it provided the complete computation for object tracking [47]. The GMM was implemented as the main tracking algorithm with a morphological and a median filter applied in order to remove unwanted noise in the image [47]. However, in the frames in which the object tracking was incomplete the optical flow method was able to support the main GMM algorithm [47]. Chauhan and Krishan used MATLAB 12 and C++ as the main software development tools. Chauhan and Krishan results suggested that combining these two methods leaves room for improvement as it the optical flow method did not catch all the untracked frames from the GMM, meaning that a more accurate follow-up method such as a background subtraction method would be possible.

Poiesi, Daniyal and Cavallaro (2010) followed a slightly different approach implementing motion flow (see section 3.1.6) instead of optical flow as used in Chauhan and Krishan's research. The motion flow algorithm was applied to estimate the regions of convergence for a football player and locations of a football [48]. The method consisted of two-step process. First, the player regions on the image were estimated by evaluating the points where the optical flow algorithm met [48]. Then the players were temporarily ignored to acquire an accurate estimation of the balls location over time [48]. To identify the ball position the same method was applied as to identify the players, the meeting points of the motion flow were evaluated giving the balls location. To find the meeting points of motion flow, the motion vectors of two sequential frames were subtracted [48]. The results showed that the ball detection has a success rate of 82% and further suggested that motion-based detection of the players is more accurate when the players are running towards the ball, as the estimate was assigned to the player position and not the point of focus [48]. A further suggestions to increase the success rate as to integrate perspective view cameras as this will allow a more detailed analysis of the trajectories of the player.

3.2.9 Object detection and calculating velocity using optical flow

Using optical flow for vision based object tracking was the main aim of Luo, Mullen and Wessell's (1988) study. The main accomplishment of Luo et al. was showing that the motion of a ridged object could be identified through optical flow, which would better facilitate real-time robotic tracking. However, only one object was tracked at a time and the object was undergoing a one-dimensional translation with no change in direction, meaning that Luo et al. study would not experience any occlusion issues faced in the previously mentioned studies [49]. By tracking only one object in an one-dimensional translation Luo et al. could track the object through consecutive frames by recognising specific pixels from the object and looking for those pixels in the following frames [49]. Additionally, optical flow was implemented to identify the velocity of the moving object [49]. Using two consecutive frames and locating the centre of the object in each frame allowed identifying the distance travelled by the object, giving the necessary information to calculate the velocity of the object. [49]. The average error between the actual recorded velocity of the object and the calculated velocity was 10% due to image noise. Where as the overall success rate of the object tracking was 87%. To ensure the validity of the test different conditions were tested, meaning under different light conditions and different shapes/objects including a golf ball.

Similarly to Djekoune et al. and, Chauhan and Krishan, in Castillo, Beltrán and Zannatha's (2011) study C++ was combined with the OpenCV library in order to implement an effective calculation of optical flow in real time. Their aim was to identify the speed of an orange football, which is more easily identified, from the background then a white football [50]. Castillo et al. applied the optical flow as it defines the velocity field in the image, which can change between frames (see Figure 12) [50].



a) Ball direction b) Velocity calculation

Figure 12: Velocity of ball calculated through optical flow [50, Fig 9]

The algorithm, applied in Castillo et al. study was based on the Lucas-Kanade method to estimate the speed of a ball. Castillo et al. chose this algorithm due to its easy implementation, as well as computational requirements being relatively low compared to other tracking methods such as the HCT [50]. Two frames from the video were extracted and an image for each frame was created. Using the OpenCV functions "allocateOnDemand" and "cvConvertImage" both images were converted into a single channel grey scale image [50]. A corners extract OpenCV algorithm was introduced "cvGoodFeaturesToTrack", to compute the second derivative [50] returning a list of points, which were used to track the football. After successfully applying these features the OpenCV function "cvCalcOpticalFlowPyrLK" was applied, which runs the Lucas-Kanade algorithm [50]. Once these functions were complete the optical flow direction of the football was plotted providing an approximate measure of speed and direction in an (x,y) mathematical format [50]. Results from Castillo et al. study included accurate ball tracking and suggesting that a coloured football helps detect the ball easier. Coming to a similar conclusion to Pingali et al. study, who identified that due to the colour of a tennis ball (yellow) the ball could be easily detected as the contrast between the ball and the background was higher [30].

3.2.10 Link to project

In conclusion, the reviewed literature demonstrate several advantages and disadvantages regarding object detection and tracking. These advantages and disadvantages were evaluated and integrated into the Putt Training Tool. As reviled through the literature the OpenCV (operating in C++ and python) and MatLAB (operating in C/C++, Java and Python) libraries hold a high compatibility with object detection and tracking methods [12]. Therefore, a decision regarding which library fits best was a key stage during software development. Once the OpenCV library is decided the development language C++ was chosen [12].

As suggested in the study of Wang et al. frame-to frame tracking through HCT is suited for individual or one-on-one sport such as golf. Therefore, golf putting would be ideal as no occlusion issues could occur. However, the frame images needed to start about 10cm in front of the player, to ensure the putter head does not hide the ball during the first few frames. Although, when considering Mai et al. study the HCT should be exchanged with the ellipse detector, as due to shape changes the ball would be identified more easily. Additionally, Seo et al. study introduced histogram calculations to identify a moving object. As the majority of this project holds only green grass in the frame this seemed like a tracking option. Additionally, the white ball was easily identified against the green grass. However, problems occurred due to shadows in the outside environment. As mentioned in Tong et al. (2010) study the direction of illumination plays a deciding factor on whether this tracking method fits the expected criteria. If however shadows do cause problems, the image invariant method from D'Orazio et al. study could be applied. Furthermore, background subtraction was tested, as additional movement in the frame was be limited. Pingali et al. study mentioned that this method works well for tracking singly moving objects. Due to the nature of putting the ball does not move in extreme speeds, therefore Pingali et al., Tong et al. and Gomez et al. issues were irrelevant. However, a fairly slow moving object could create issues for a background subtraction algorithm. Meaning the ball does not hold a new set of pixels for each frame. Archana and Geetha study pointed out that the background subtraction algorithm could then detect half of the ball as a non-moving object.

By comparing different studies it became clear, that the only way to identify which method suits the given environment was to test all the methods under the same conditions, achieving an accurate comparison of the methods. As Mao et al. suggested a combination of two techniques helps to exploit the strength of each technique.

As Barratt, Deterding et al., and Lee and Cho identified, gamification within sport application to be successful, this research project implements similar techniques such as high scores and award systems. Additionally, a visual representation of the data was provided, where users could track their progress within different fields:

- Track the distance the ball is away from the hole (award stars)
- Amount of putts holed (high score)

Introducing a game-like element improves performance over time as well as motivate the user to beat their personal putting high score and collect as many stars as possible.

4 Project Implementation

The final product of this project is the foundation for a golf-putting tool, which acts as a learning aid for amateur golfers, combining putting with gamification to motivate users and improve putting skills. Computer vision techniques were applied to collect accurate ball-tracking data, which was then visualised in a graph format where high scores are recorded and stars awarded to demonstrate the potential for gamification.

All development was undertaken on a Mac OS X 10.9 operating system, however the OpenCV software utilised was chosen in part due to its compatibility with Mac, Windows, and Linux operating systems.

This section will discuss which techniques were chosen to create an accurate golf-putting tool. Different techniques were tested under the same conditions to justify the choices made. This section is split into three main computer vision sections of the project :

- Development decisions
- Ball tracking
- Identify slope through deceleration

Each section gives a detailed break down on the decision process, which was faced.

4.1 Development Decisions

For the main ball tracking analysis the OpenCV library was chosen over MATLAB, as OpenCV requires a lot less memory storage. As OpenCV can be implemented in different development languages a decision between python and C++ was left. C++ was chosen, as python is a wrapper around the original C++ code. Additionally, previous experience in C was used as an introduction to C++. However, it was important to identify the advantages and disadvantages of OpenCV as a library, which is shown in Table 1:

Table 1: Advantages and disadvantages of OpenCV as a library

Advantages	**Disadvantages**
OpenCV allows changes and adjustments of functions within the library depending how they would best fit your project	**OpenCV's documentation of some functions is weak and therefore difficult to understand exactly what it is doing**
OpenCV holds a fast computation time which is beneficial for image processing	**As many C++ tools the visualization and debugging of the tool shows to be difficult**

To graphically display data a choice between Java and HTML5 was made. HTML seemed to be the best choice, as it is a cross platform language. Additionally, combining HTML with JavaScript allowed interactive functionality. Changing the graphical display in response to events. Further, for the JavaScript no additional graphing library such as Chart.js was chosen to allow more freedom within the creation of the graphs as well as the ability to easily adjust the graphs for future work.

The main code holding the ball tracking and deceleration calculations was 706 lines long. In this code two computer vision techniques were present, background subtraction and ellipse detection. This code also consisted of all the supporting elements such as the perspective transform and distance calculations. The HTML and JavaScript code was slightly shorter with a total of 272 lines (consisted of 24 lines of HTML and 248 lines of JavaScript). The JavaScript file was longer than the HTML because the HTML only calls the events where as the JavaScript act upon the events, for example taking data and creating the appropriate graph and identifying the high score.

Maintaining clean and clear code is key throughout the duration of this project. It was important that the code can be easily to understood and modified. Additionally, it was seen as very important that changes within one area of the code did not affect the rest of the code. For instance if the ellipse detection element was adjusted the ball tracking was not effected. For future work the code was also well documented and commented to enhance the maintenance of the code. In order to avoid bugs and errors being introduced into the main code base, development was conducted in stages. Where possible, sections of the program were developed in isolation and tested before being integrated into the main program. The advantage of this approach was that it maintained a clean and fully functioning master branch of the project at all times.

On this last point, Git was implemented to enable version control and to ensure that the project could be returned to a previous state if bugs were accidentally introduced into the main code base.

4.2 Ball Tracking

This section details the implementation of the following project objectives (see section 1.2):

- Identify and implement a suitable technique for tracking the golf ball
- Develop software which enables the application to measure the distance between the golf ball and the hole

In order to achieve these objectives, the ball tracking techniques discussed in section 3 were implemented and compared in order to find the most suitable for this application.

4.2.1 Ground Truth

To evaluate which technique was best suited to tracking the golf ball, each technique was implemented in OpenCV and then the resulting estimates of the ball position were compared to the ground truth (the actual position of the ball). The techniques implemented were:

- Ellipse detection
- Background subtraction
- Colour recognition
- Hough Circle Transform
- Optical flow

This was done on a frame-by-frame basis for 102 videos to ensure a statistically significant sample size was used. Data was outputted into a text file for further comparisons. The reason behind storing data in text files was due to the simplicity of text files as well as being easily adjusted and integrated into different programs such as Excel. Appendix A shows the difference between the ground truth and each tracking technique's estimate of the ball position for each frame of video 32. It can be seen that the optical flow and Hough Circle Transform techniques often tracked objects other than the ball, reducing their accuracy. Table 2 displays the average distance from the ground truth for each of the techniques across the 102 videos used in this study.

Table 2: Average difference from the ground truth for each tracking technique studied

Distance from Ground Truth (cm)					
	Ellipse detector	**Background subtraction**	**Colour recognition**	**Optical flow**	**Hough Circle Transform**
Average	4.9	4.5	8.7	10.4	30.1

Table 2 illustrates that the ellipse detection and background subtraction techniques produced significantly more accurate estimates of the ball's position than the others. This data was collected on six different putting greens where different conditions were employed to expose the strengths and weaknesses of each tracking technique. These conditions included:

- Still shadows in the frame

- Moving shadows in the frame
- Moving objects in the frame
- Different ball colours
- Different putt lengths

The next two sections cover the background subtraction and ellipse detection implementations in more detail.

4.2.2 Background Subtraction

Background subtraction overall produced the most accurate results for ball tracking. However, conditions where background subtraction did not perform accurate included moving shadows and objects in the frame. Exposing the weaknesses to this technique. Nevertheless, due to its high accuracy in performance for still shadows, changing ball colour and changing lengths background subtraction proved to be the most suited. Additionally, in real life situations the frames will not contain any moving shadows or objects, just the ball and the green as shown in Figure 13. Therefore, reducing background subtraction's error margin.

Figure 13: Typical frame for ball tracking

Once the OpenCV background subtraction filter was applied to the frame, only the ball remains and the rest of the image is black (see Figure 14). However, as discussed above, if a moving object enters the frame then errors can be introduced which interfere with the tracking (see Figure 15).

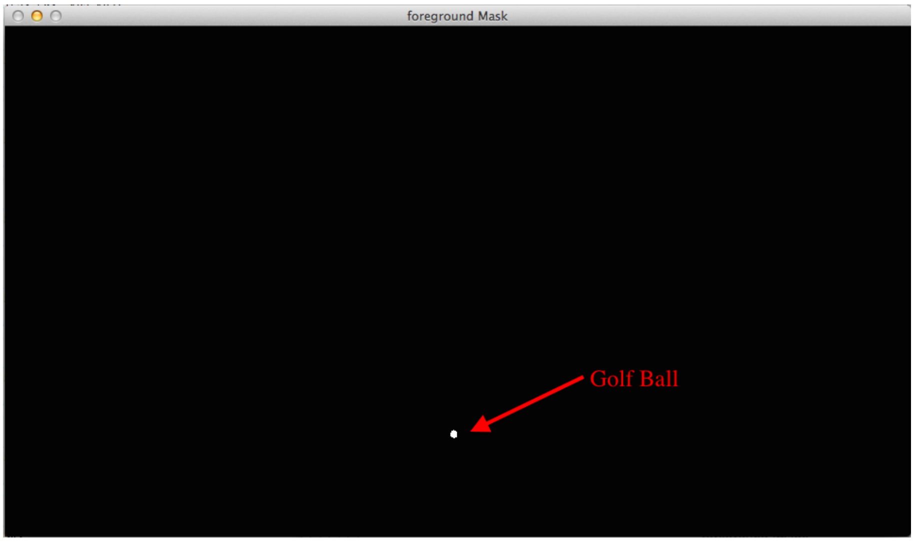


Figure 14: Example application of the background subtraction filter to a frame with no additional moving objects (red label and arrow added for clarity).

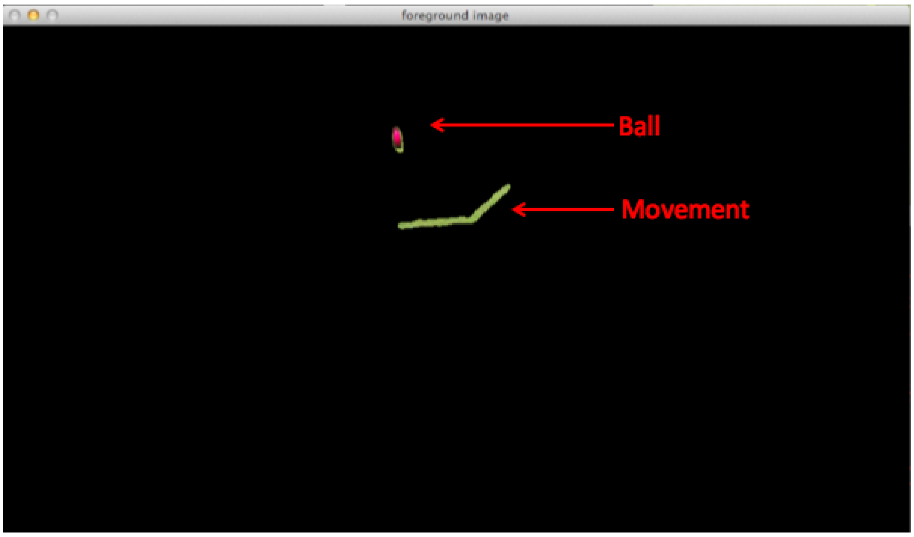


Figure 15: Example showing the noise introduced when the background subtraction filter is applied to a frame with additional moving objects (red labels added for clarity)

Therefore, for the following development stages of the application, videos with no additional moving objects on the green were using for training and testing. Ball tracking is seen to be a difficult task due to the balls high speeds and constant size changes from one frame to the next. Therefore, basing a ball tracking method on its size, shape or colour gives erroneous results. Given that background subtraction is unable to identify stationary objects, a second detection method was required to identify the hole in each frame, which is discussed in the next section.

4.2.3 Ellipse Detection

In terms of identifying the hole, ellipse detection produced more accurate results than the Hough Circle Transform after testing both techniques on 102 videos. This was due to the hole not appearing to be perfectly round in a portion of the videos depending on the illumination. Figure 16 shows how the hole is detected after the ellipse detection filter is applied.

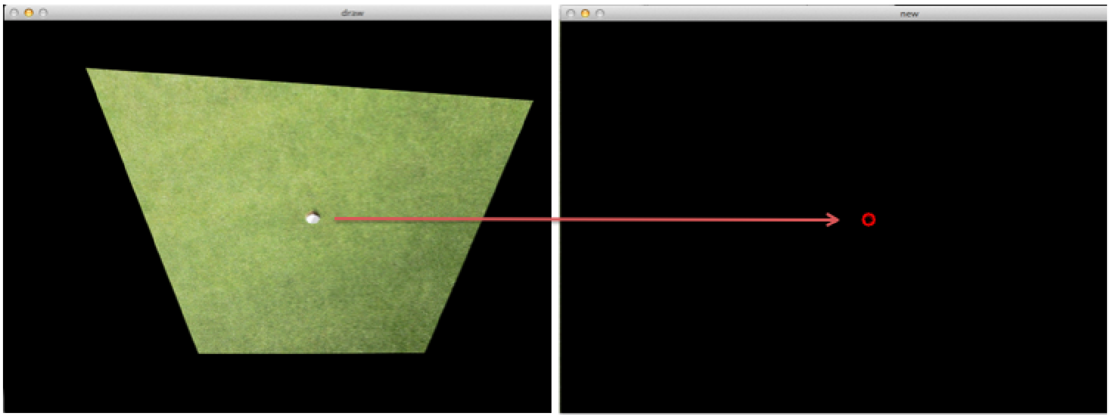


Figure 16: Example of ellipse detection being used to detect the hole

Using two different techniques, one to track the ball and the other to detect the hole, ensured that the ball and hole could not be mixed up or confused. When the Hough Circle Transform was tested, there were at least two circles detected in each frame, the hole and the ball. When the ball got close to the hole there would then be confusion within the data structure, leading to erroneous results. Two methods were therefore employed to eliminate the scenario of the ball and hole being mistaken each other.

4.2.4 Perspective Transform

In order to be able to measure the distance from the ball to the hole, each video's perspective needed adjusting to give a bird's eye perspective. A reference square with a known size was placed in the middle of the frame as shown in Figure 17. A transformation matrix was then used to transform the square back into a perfect square as shown in Figure 18. This enabled the pixel size to be determined (as the dimensions of the reference square were known), which meant the distance could be calculated using equation 8:

$$d = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \tag{8}$$

Figure 17: Reference Square before perspective transform

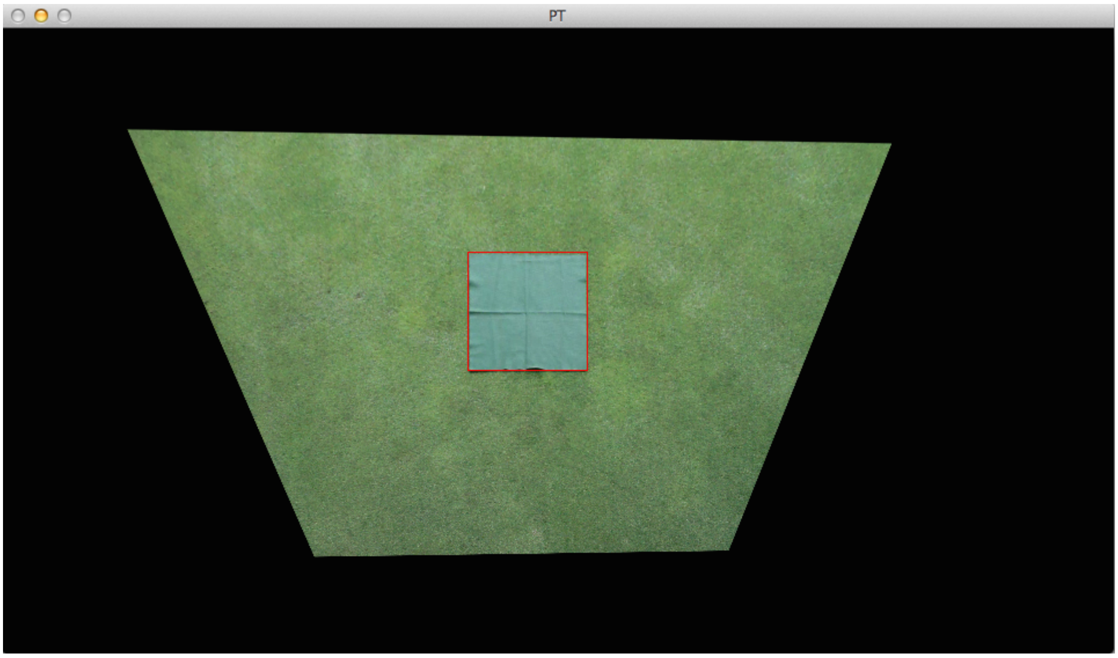


Figure 18: Reference Square after perspective transform

As the videos were recorded, the distance between the ball and hole was measured and documented. Therefore, it was trivial to test whether the program's perspective transform enabled distances to be accurately calculated from the frames. The program was tested on videos that featured different length putts (1m and 2m between the camera and hole) and varying breaks of the green (uphill, downhill, flat).

Appendix B compares the distance calculated by the program to the actual distance measured on the greens. Figure 19 shows that the further the ball is from the hole after the putt has been taken, the greater the error in the computed distance.

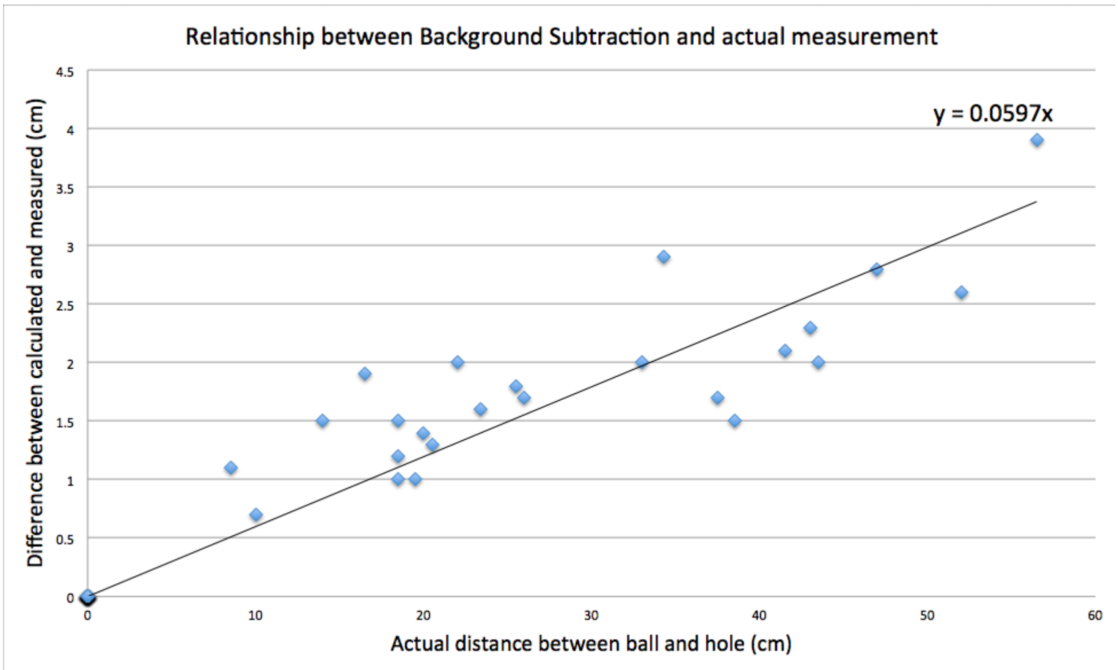


Figure 19: A visualisation of a 1-meter flat putt showing the relationship between background subtraction and actual measurement

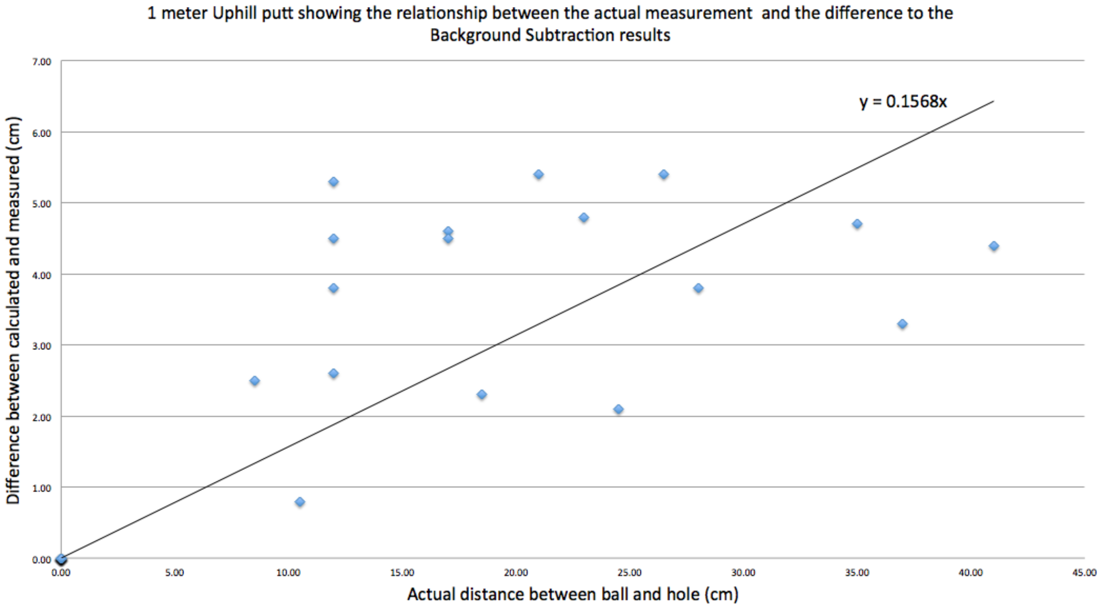


Figure 20: 1-meter uphill putt illustrating the difference in measurement between background subtraction and actual measurement

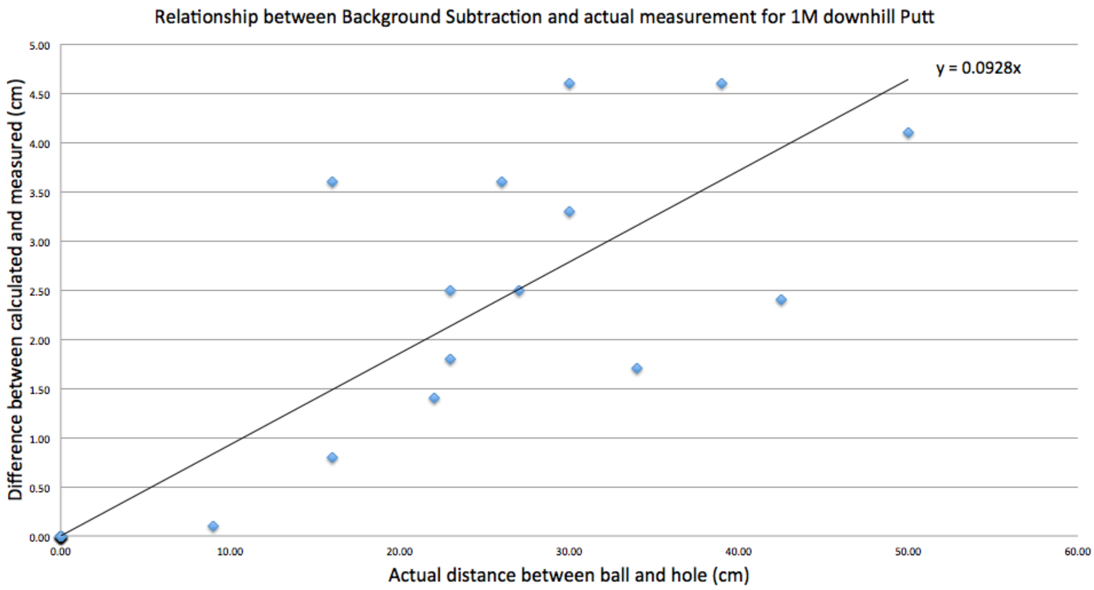


Figure 21: A visualisation of a 1-meter downhill put which shows the coloration between distance the ball is from the hole and the difference between background subtraction and actual measurement

Figure 20 (uphill putt) and 21 (downhill putt) further support the argument that slope is the reason of the calculations to become more inaccurate the further the ball is from the hole. The trend lines gradient for the uphill putt (0.1568) and the downhill putt (0.0928) hold a greater value then for the flat putt (0.0597).

Meaning that the uphill and downhill slope causes a greater error margin. Showing that the putts performed on the flat slope hold the smallest error. Indicating that the slope of the green affects the accuracy of the measurements therefore should be taken into consideration during the perspective transform.

4.2.5 Discussion on Ball Tracking

Despite the ball tracking techniques being tested in multiple environments, there were several issues that were faced which require consideration given that golf is an outdoor sport. Shadows were sometimes introduced to the frame by nearby moving objects, which interfered with the ball tracking method and introduced significant noise into the data set. Adjusting the colour space to remove the shadows could potentially have eliminated this issue, and indeed such a method would represent a possible extension for this project. However, given that one can easily avoid these shadows on a typical green, the issue was considered a lower priority than enabling the adjustment of the perspective transform depending on the slope.

4.3 Identifying the Slope through Deceleration

This section details the implementation of the following project objectives (see section 1.2):

- The deceleration of the ball can be accurately computed
- The slope around the hole can be determined and visualised

In order to achieve these objectives, videos featuring varying slopes were tested under the assumption that putts on flat ground would display constant deceleration of the ball, while those on uphill and downhill slopes would display greater and lower deceleration respectively. Additionally, if the deceleration could be accurately mapped then the perspective transform could be proven to be accurate.

4.3.1 Mapping Deceleration

Once the deceleration at each ball position in the video was calculated, it was entered into an array in an attempt to visualise the slope of the green. An array was created for each set of videos (with the same camera perspective and distance to the hole), which contained a visualisation of their overall deceleration data. Figures 22 visually shows the result for the 1m downhill putts where the red squares represent an uphill and the green downhill, the hole is represented as a red circle for clarity as to its position. Appendix C, D, and E show the exact numerical result split into three segments of the 1m downhill putt. In Appendix D, the hole is represented as a red circle for clarity as to its position.

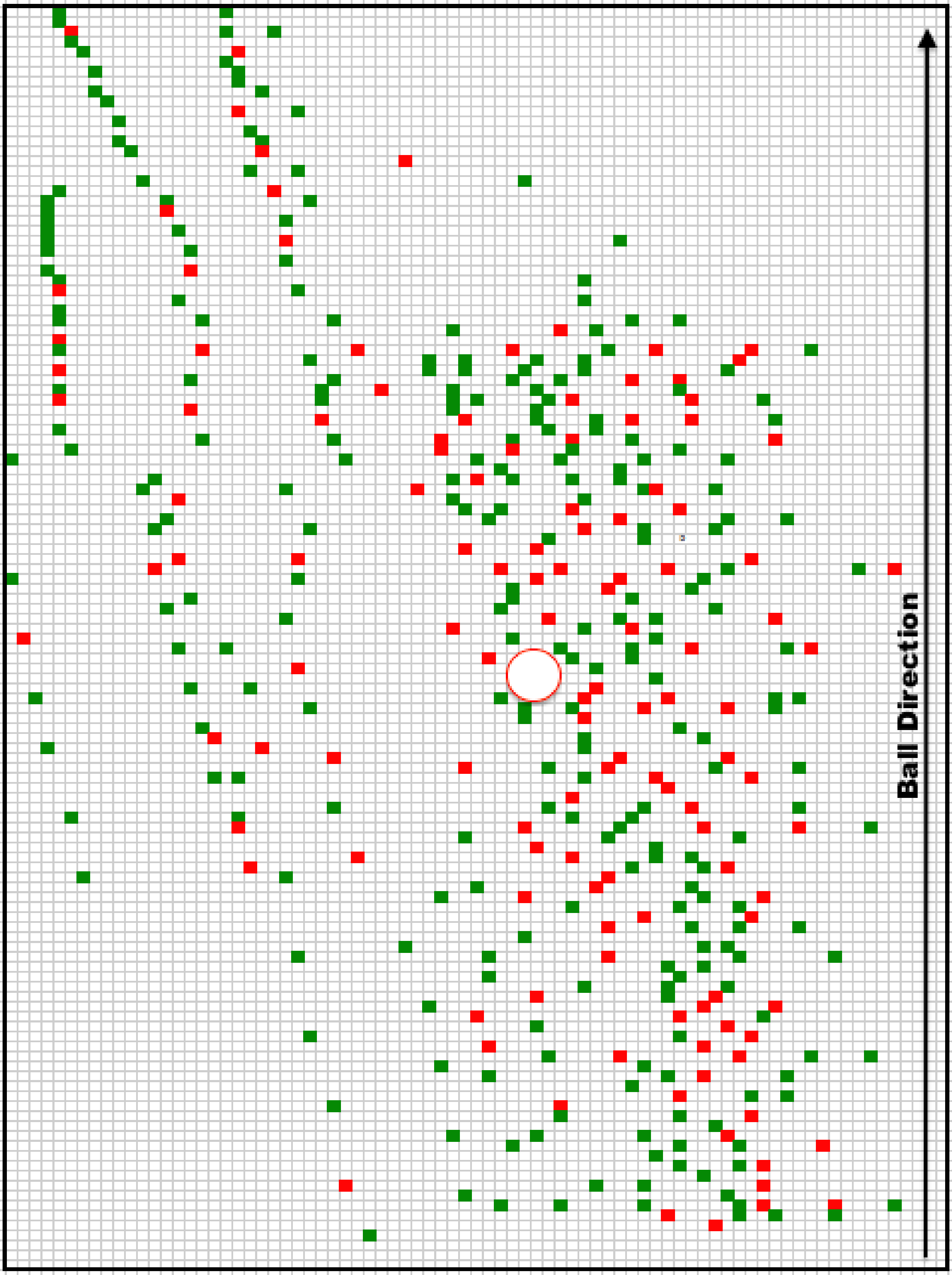


Figure 22: Deceleration map of the golf ball of the 1m downhill putt

The map produced by this method showed that the ball was decelerating but it was not clear whether the slope was going uphill or downhill. Therefore graphs displaying the velocity of the ball in each frame were produced (see Figure 23).

As this figure shows, those graphs did not display a smooth decline in velocity but instead a step-wise decline due to noise. Given the videos were taken in an outdoor environment, there are a number of potential causes of this noise including the grass growing against the direction of the ball (which could increase the rate of deceleration), wind acting upon the ball, or frame inconsistency. Therefore further graphs were produced displaying the averaged velocity in each frame across 50 videos for 1m (Figure 24) and 2m (Figure 25) putts on the same slope.

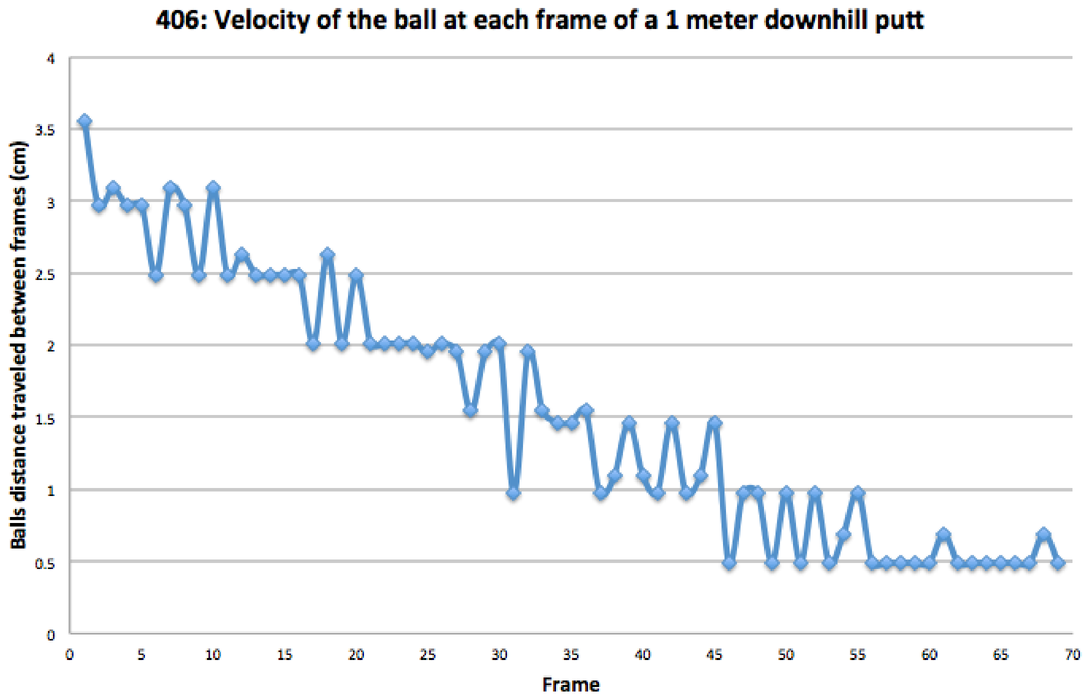


Figure 23: Velocity of the golf ball during a 1m downhill putt (video 406)

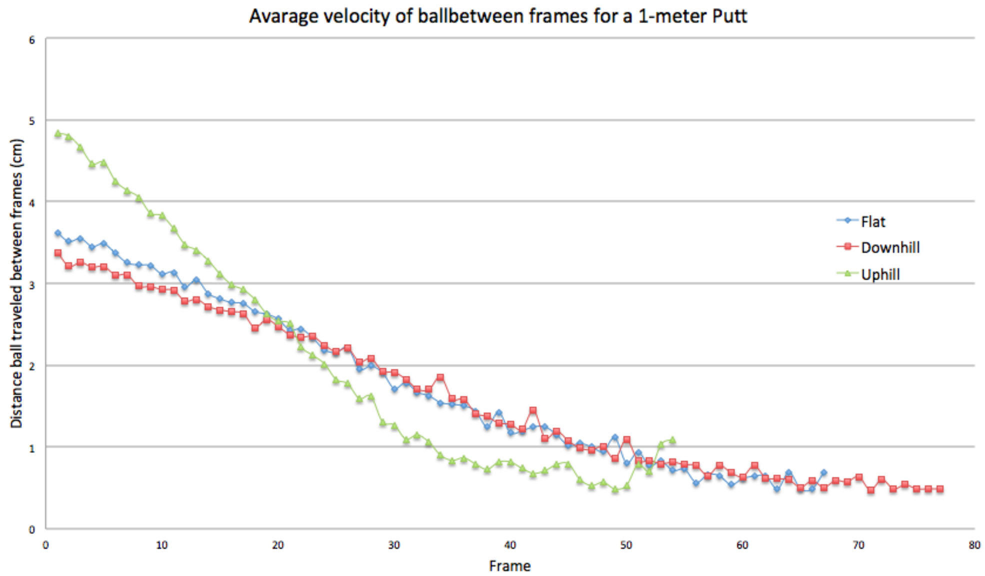


Figure 24: Velocity of the golf ball averaged over 50 1m putts for flat, downhill, and uphill slopes

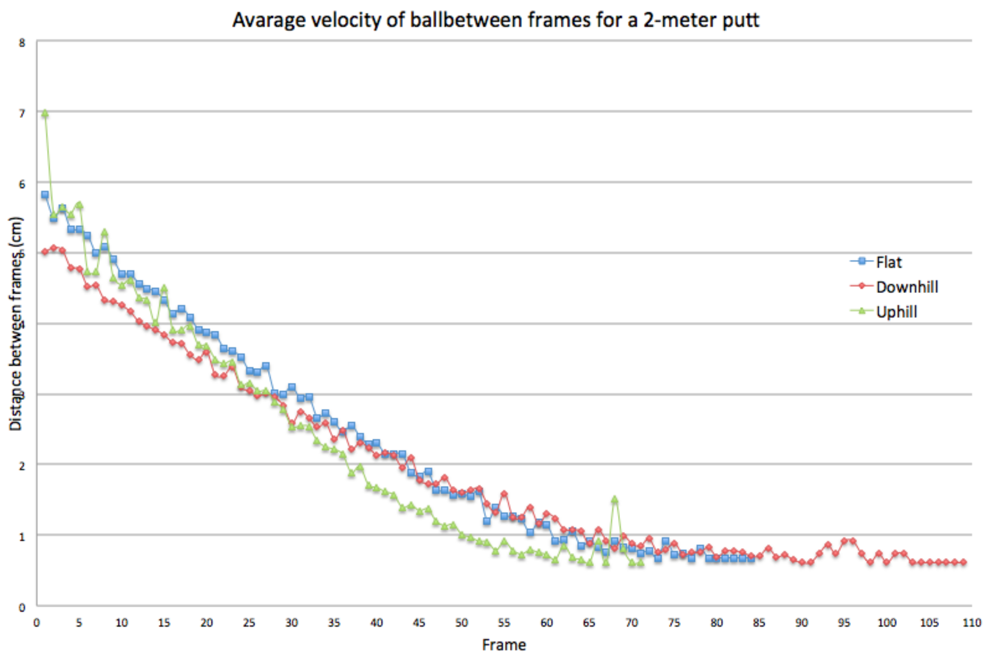


Figure 25: Velocity of the golf ball averaged over 50 2m putts for flat, downhill, and uphill slopes

Both the 1m and 2m graphs (Figures 24 and 25) demonstrate that averaging the velocity over 50 videos significantly reduces the level of noise, leading to smoother curves. Each graph contains three curves representing flat, downhill, and uphill putts. The flat putts demonstrate constant deceleration initially before the rate drops off as the velocity tends towards zero. The downhill putts exhibit a lower rate of deceleration in comparison, with the ball in motion for a longer period of time despite a lower average initial velocity. The uphill slope exhibits the steepest curve in both graphs, with a significantly higher rate of deceleration than both the flat or downhill putts and a shorter period in motion. Note that the momentary spike in velocity of the ball in the uphill putts is due to the ball beginning to roll back down the slope.

4.3.2 User Practicality

The high number of videos required to reduce the noise in the velocity graphs means that this method would be impractical for users. Most golfers would target 10-20 putts per exercise. While the information gathered through mapping the green presents some interesting results, this method was not suitable for the purposes of this project.

While the method was not satisfactory, valuable results were gained from the above exercise. The expectations regarding the deceleration of the golf ball on different slopes were proven to be correct through the averaging of a high number of videos. This fact also demonstrated that if the amount of external noise through the environment could somehow be reduced, then the velocity graphs could prove a useful tool to help improve a user's putting skills. Additionally, the objectives set for this task were met as the ball's deceleration in each frame was identified and used to map the slope of the area around the hole. However, the second objective was only met through averaging the results of a large number of videos.

5 Data Visualisation

In this section the following objective from section 1.2 was targeted.

- Create gamification through a graphical display of the data

HTML5 and JavaScript were used in combination to visualise the data due to their cross platform compatibility. The axes of the graphs adjust automatically according to how many weeks of data have been recorded, presenting the user with the best possible visualisation. Line graphs were chosen for this visualisation as they enable the clearest display of the putting data and clearly demonstrate trends in a user's performance. This enables the user to check their progress over time and see where they have made improvements and where weaknesses still remain. By enabling the user to visualise their data in a graphical form, an element of gamification was introduced as users can gain extra incentive from being able to track their progress as well as beating their previous high score .

5.1 Gamification through graphical displays

Increasing user motivation through game-like elements is key in gamification. For this learning tool creating the competitive environment acts as the game-like element, which provides user motivation. As each week's putting distances are compared to previous weeks, where the user is stimulated to improve their personal best score. Figure 26 shows the average distance between the ball and the hole for each week. If the current week holds a lower distance compared to the previous week a star is reworded. This reword system creates motivation for the user and feeling of accomplishment creating another game-like element. The data in Figure 26 shows a constant downward trend awarding one stars for each week, where as Figure 27 has a higher average for week four therefore does not receiving a star for that week.

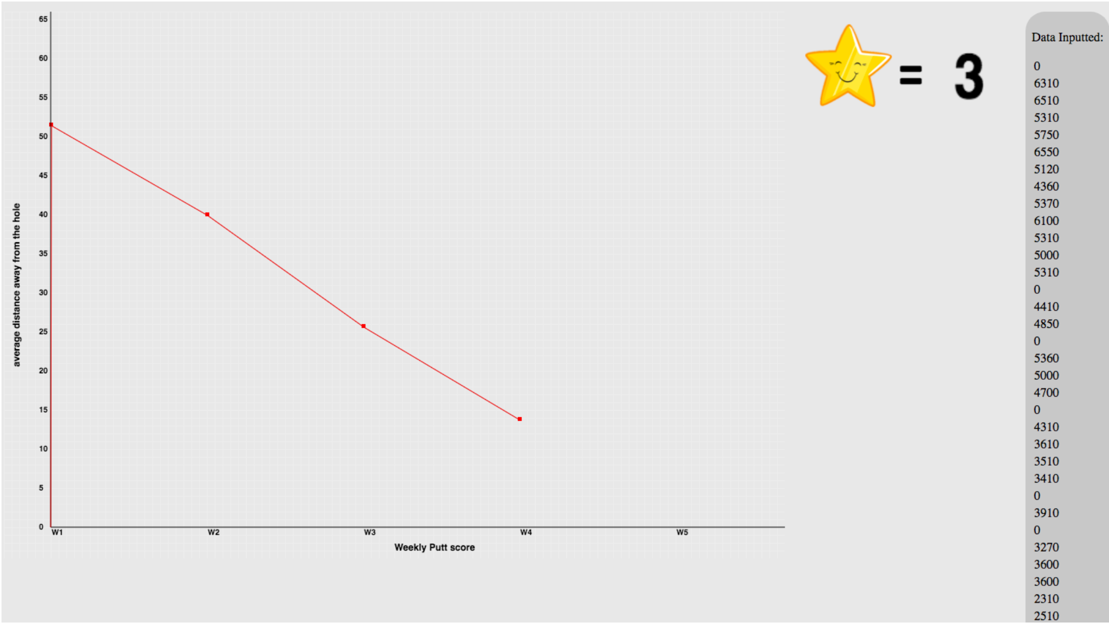


Figure 26: Four weeks successful star award system for 2m putt

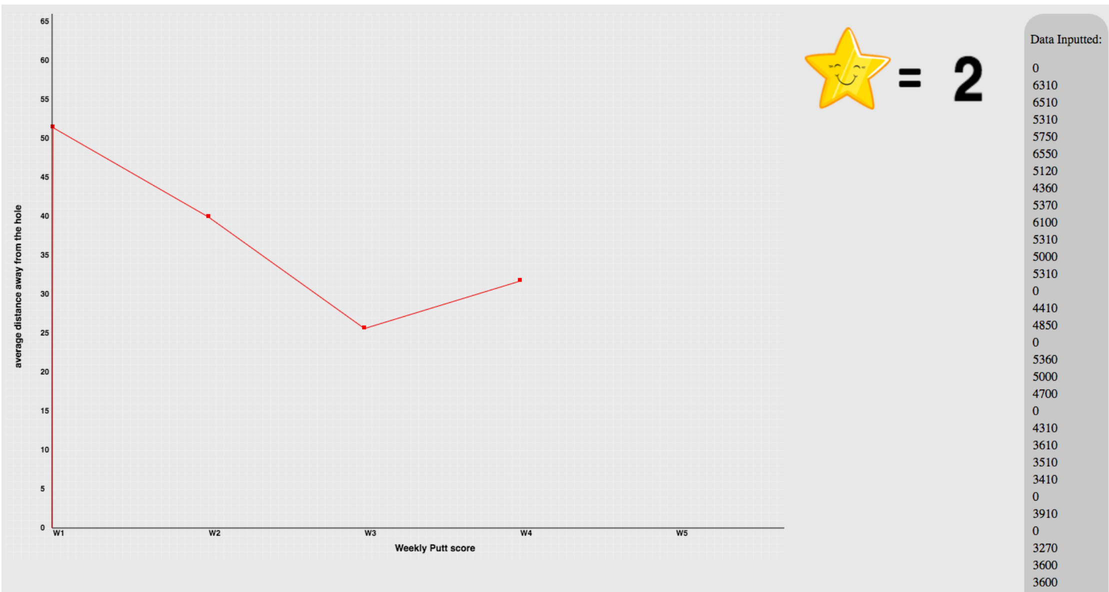


Figure 27: Four weeks unsuccessful star award system for 2m putt

Another gamified statistic, which the user can track and visualise, is the amount of putts holed each week. Figure 28 shows a continuous increase for 1m putts holed and Figure 29 shows 1m putts holed with setback at week four. A high score tool is introduced for the user to identify in which week the highest number of putts were holed and the number of holed putts. Additionally, creating an incentive for the user to aim to beat the overall high score.

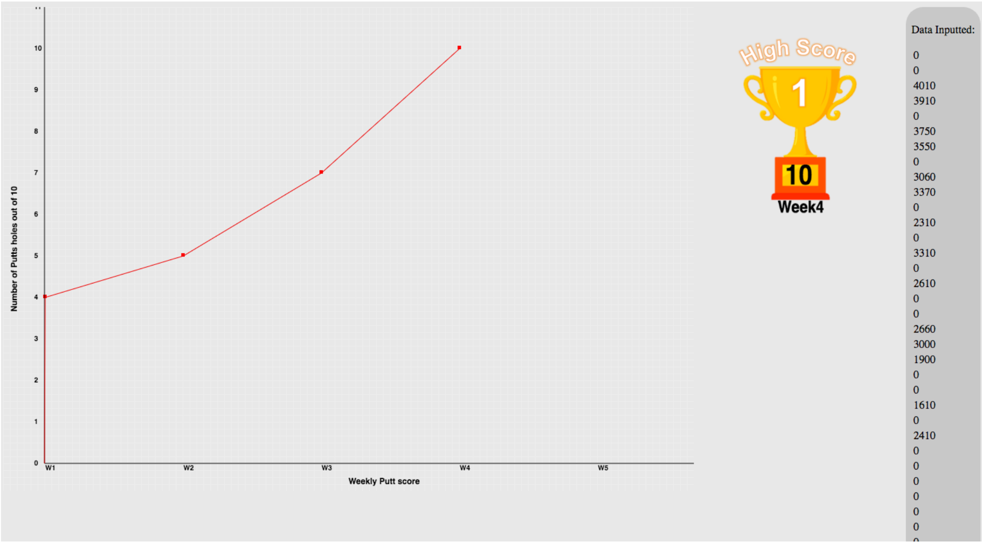


Figure 28: Continuous improvement for 1m putts holed

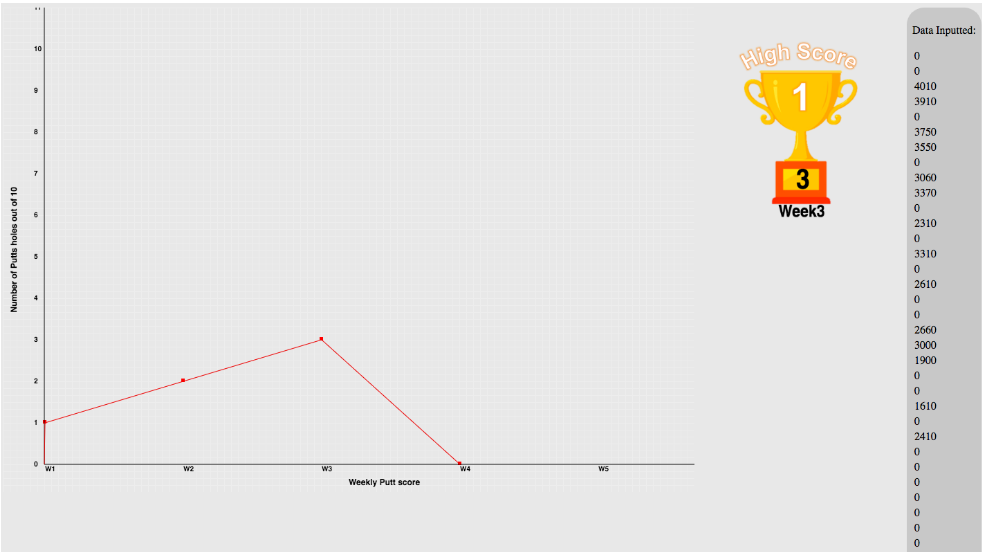


Figure 29: 1m putts holed with setback

5.2 Player progress

As section 6.1 averaged the length between the ball and hole for each week there is an additional option, which lets the user plot each putt within the time frame. This enables the user to see whether the average is an accurate representation or whether each week one putt extremely lifts the average. Figure 30 shows the distance the ball was from the hole for 1m putts over three weeks.

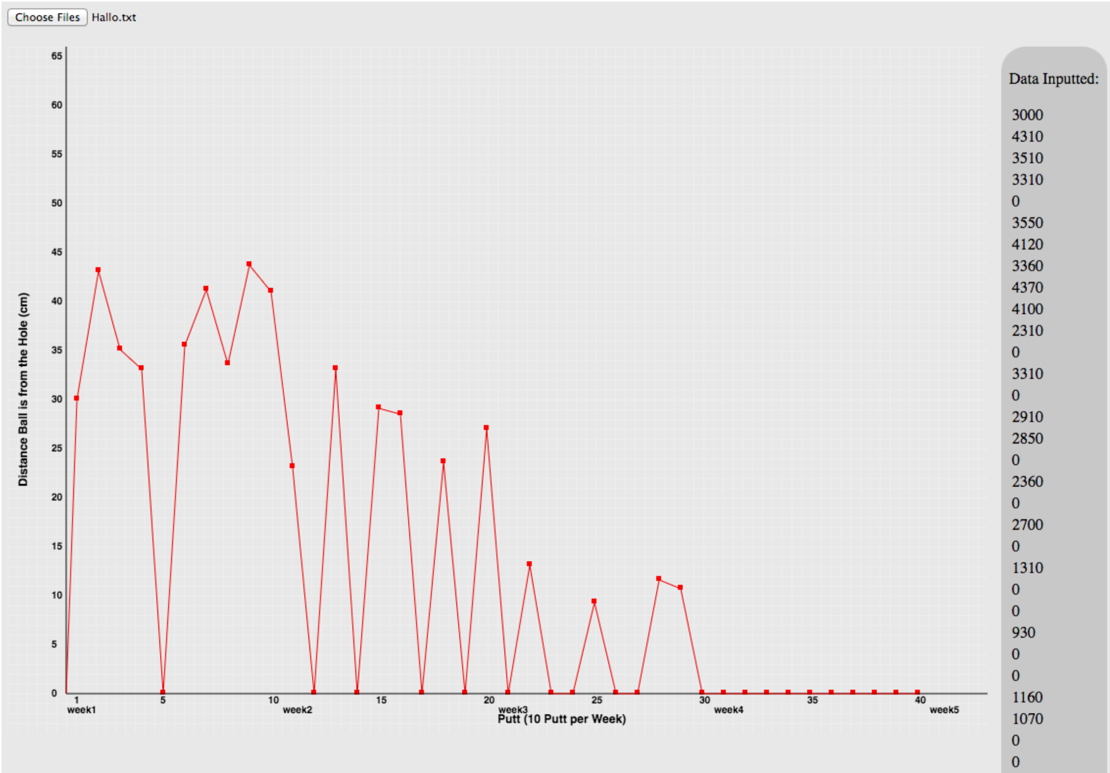


Figure 30: Progress of 1m putts over three weeks

Through visualising their progress, the users can clearly identify where putting weaknesses lie. This allows the user to deliberately place more emphasis on strengthening those areas. Figure 31 shows the improvement of 2m putts.

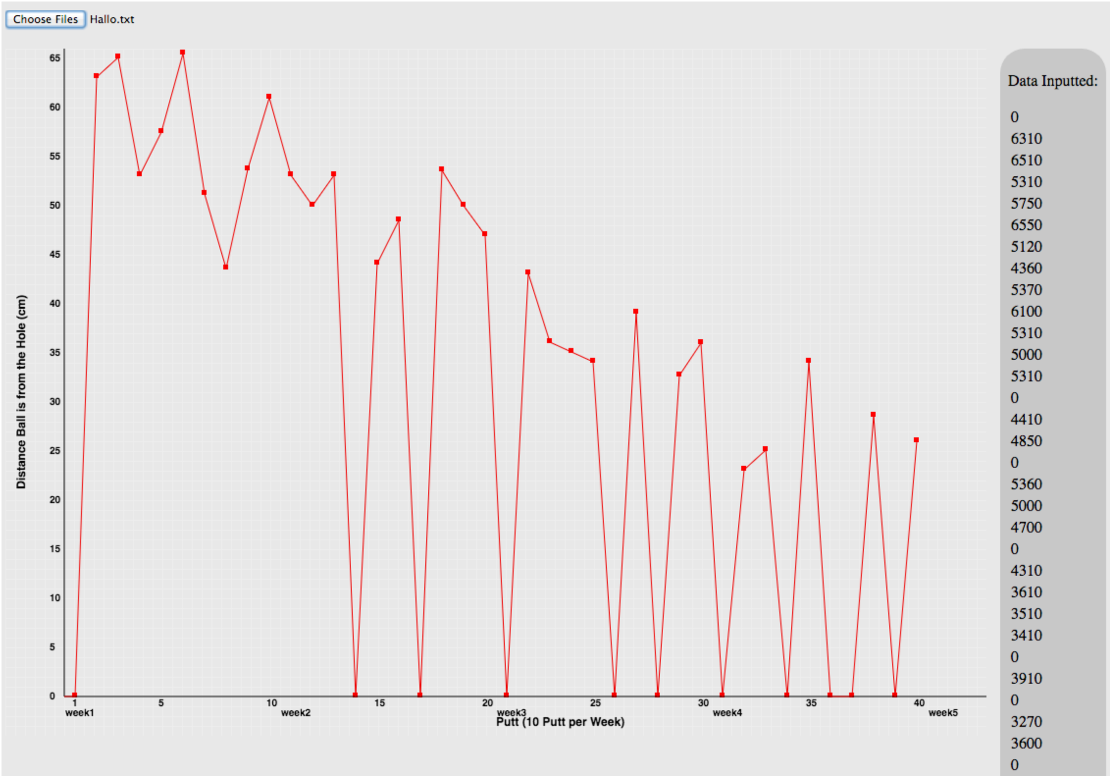


Figure 31: Progress of 2m putts over four weeks

5.3 Further work regarding the graphs

These graphs could be expanded so that different exercises are shown on different graphs. This would enable the user to practice all possible aspects of putting with instant visual feedback. A further adaptation could be a comparison between two players, identifying who is better at each exercise. This could increase each user's motivation to improve by creating competition through gamification. Additionally, further work could involve motivating users through social media allowing the user to post and share the number of stars earned or the current high score.

6 Results and Evaluation

Initially this section will discuss the functionality of the golf putting tool, before moving on to a critical evaluation of the tool's performance. To ensure reliable results, 520 videos were tested which were each filmed at 25 frames per second. To ensure that the results could be compared to the ground truth, measurements were taken for each video recording of the distance between the ball and the hole. This enabled the tool's output to be compared with the actual values.

6.1 Evaluation of the Tool's Functionality

Section 4.2 demonstrates that the golf-putting tool can successfully detect and track the golf ball in a variety of situations, including different putt lengths and on greens with varying slopes. As described in Section 4.2.1, the tool's output was compared with the ground truth measurements recorded alongside each video to ensure the tracking was accurate. Additional functionality was covered in Section 4.2.4, which displays the tool's ability to measure the final distance between the ball and the hole, and to determine if the putt has been holed.

6.2 Evaluation of the Tool's Performance

To test the tool's accuracy in computing the distance between the ball and the hole, the calculated value was compared against the actual measured value. Therefore it was crucial to ensure that accurate measurements were taken during the recording of each putting video.

To determine the tool's performance in different scenarios, it was tested in the following environments:

- Different ball colours (white, yellow, and pink)
- Putting green completely in shadow
- Putting green completely devoid of shadow
- Putting green partially in non-moving shadow (within the frame)
- Putting green partially in moving shadow (within the frame)
- Varying putting green slopes (uphill, downhill, and flat putts)
- Differing putt lengths (1m, 1.5m, and 2m)
- Differing putt speeds (initial velocity of the ball)
- Different putting greens (to account for different grass types and grass cuts)

Table 3 shows under which of the above conditions the tool was successful.

Table 3: Golf putting tool success in differing conditions

Circumstance	**Successful**	**Unsuccessful**
Different ball colour	✓	
Completely in sun	✓	
Completely in shadow	✓	
Partly in moving shadow	✓	
Partly in moving shadow		✓
Additional movement		✓
Different slopes	✓	
Different lengths	✓	
Different speeds	✓	
Different greens	✓	

The tool was unsuccessful when the video frame contained moving shadows or additional movement (other than the ball) was because background subtraction works by comparing the previous frame to the current frame, and tracking objects that have changed position from one frame to the next. Therefore, if there is more than one moving object in the frame then it is no longer certain which is the moving ball and inaccuracies are introduced. To visualise this issue, Figure 32 shows two consecutive frames with no additional moving objects and Figure 33 contains sequential frames with additional movement highlighted.

Figure 32: Two consecutive frames with no additional movement

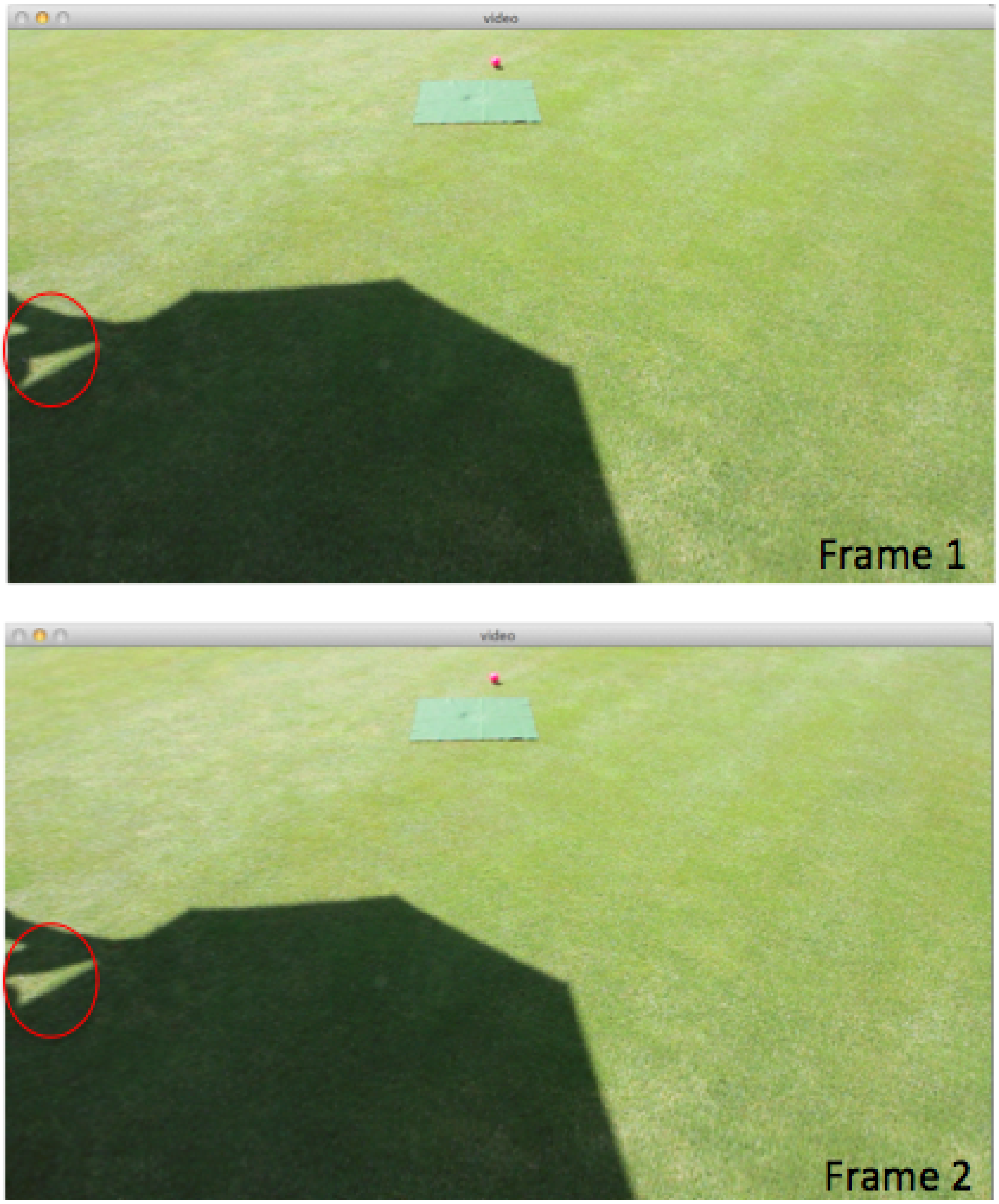


Figure 33: Two consecutive frames with additional movement

As discussed, the tool would interpret these additional movements in the frame as ball movements. This can clearly be seen from the tracking path (represented by the yellow line) in Figure 34, which has been visibly distorted by the additional movement in the frame. The ball tracking process is only successful if the path tracked solely follows the line which the golf ball took. A successfully tracked path can be seen in Figure 35. This demonstrates that additional movement in the frame is a cause of significant error for the tool, which can be avoided by preventing any moving shadows from entering the video.

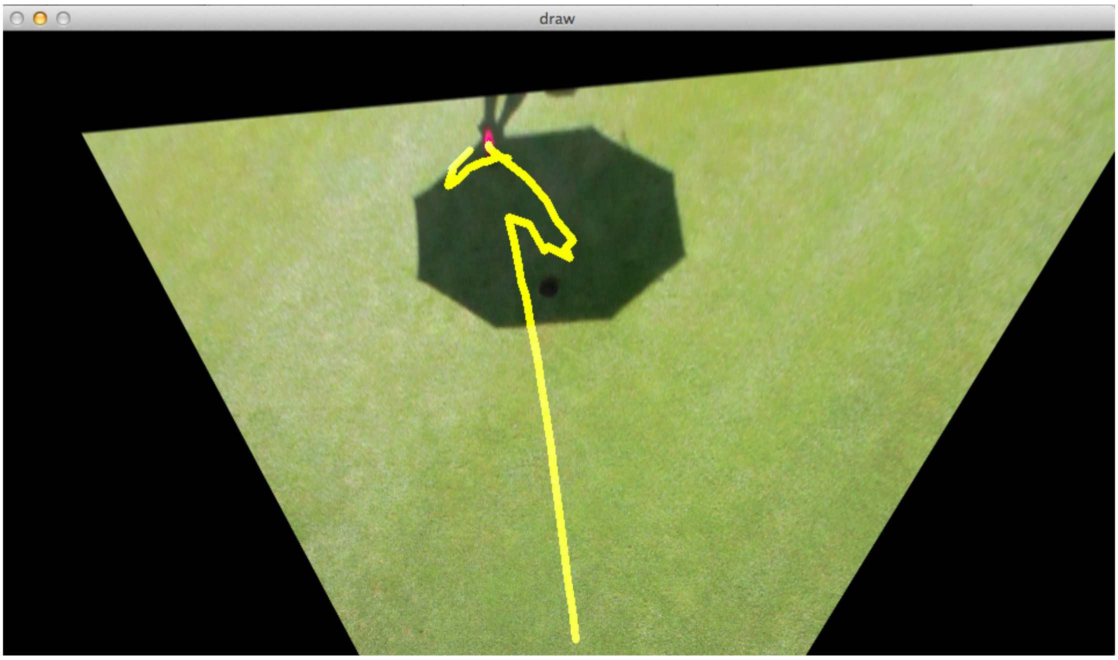


Figure 34: Erroneously tracked path of the golf ball due to additional movement in the frame of the video

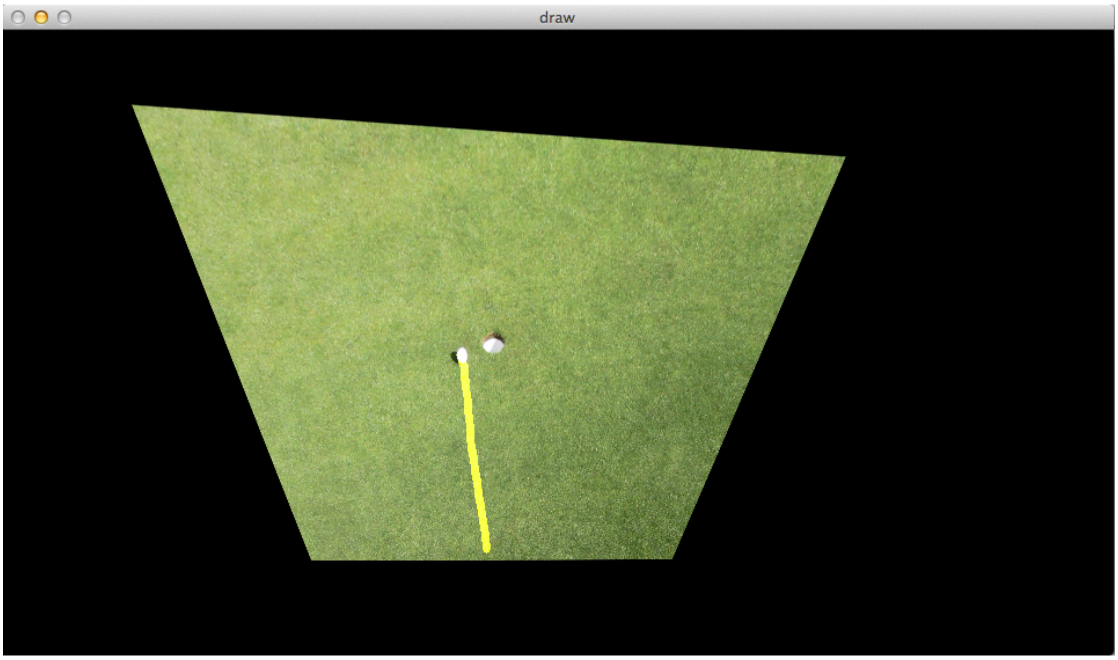


Figure 35: Accurately tracked path of the golf ball in a video

6.3 Appropriate Methods for Tracking

Following a comparison of various tracking methods, described in section 4.2.1, background subtraction was selected as the method for tracking the golf ball on the putting green and ellipse detection was chosen to detect the hole. Due to the shadows created by both the golf ball and the hole, neither could be identified as perfect circles. As background subtraction does not rely on objects being circular or a specific shape, instead identifying movement between consecutive frames, it therefore proved to be the most effective ball tracking method. Given that golf balls come in multiple different colours, and that even white balls will vary in shade depending on the light intensity, colour recognition methods proved unsuitable for ball tracking. Again, since background subtraction works regardless of the colour of the object being tracked, the user is free to use any colour of golf ball with the application.

Using background subtraction to track the golf ball also opens the tool up to various avenues of future work, including other areas of golf. For example, a user could practice their chipping by hitting multiple balls onto the green. Having multiple balls on the green would not be an issue with the background subtraction method as it only tracks the moving ball, not any stationary golf balls sitting on the green.

6.4 Potential Changes to Improve the Application

Following the application's development, several potential improvements were identified which could improve either the application's functionality or its effectiveness.

These potential improvements are discussed in this section of the thesis.

6.4.1 Multiple putts within a video

One potential improvement to the putting tool would be to allow multiple putts to take place in one video, although not simultaneously. The first ball would have to come to a rest before the next could be hit so not interfering with the background subtraction method for tracking the ball. This would allow the user to concentrate on their putting practice rather than constantly restarting the video recording for each individual putt.

This change would make the tool more user-friendly as it would be easier to use and increase the player's productivity during putting practice. Some adaptation to the ellipse detection method used to locate the hole would be required, but this should be relatively straightforward, as the hole would not move its position within the video's frame following the first putt. As the hole's position would remain constant throughout the video, it could be used as a reference point for the remaining putts tracked during the course of the video.

In addition, this modification would enable two players to practice their putting at the same time and enable them to use the application to compare their putts against each other. This would add a further gamification element to the putting tool, encouraging users to compete against one another.

6.4.2 Combine two ball tracking techniques

Another potential improvement to the golf-putting tool, which could improve its effectiveness, is to combine two different ball-tracking techniques to track the ball. Each technique would compute its estimate of the ball's location, and the average would be calculated to determine where the ball is located within each frame. If one of the tracking methods were to suddenly jump away from the previous coordinates, this could more easily be identified as noise and those erroneous coordinates could be removed from the tracking calculations. This would decrease the level of error introduced by any one ball tracking method.

Utilising two ball-tracking methods simultaneously would also enable the user to utilise the tool in environments that the existing application produces less accurate results in. Two techniques that could be combined are background subtraction and ellipse detection. Given that both have already been integrated into the application, it would only require minor changes to the code. These two techniques have been shown to be complementary, with ellipse detection unaffected by the additional movement within the video's frame, which negatively impacts the accuracy of background subtraction. Likewise, background subtraction is unaffected by multiple stationary elliptical objects within the frame, a scenario which generally leads the ellipse detection method to produce erroneous results. Therefore, these two techniques are well suited to cover each other's weaknesses.

6.5 Discussion on the Results

The implementation process of this project revealed several key findings. By utilising a high number of videos to test the tool, it became clear that external noise could be largely eliminated from the results. Integrating data visualisation methods enabled the user to track improvements in their putting over time, introducing an element of gamification. This tracking and visualisation of a user's progress over time also enables them to identify areas for improvement within their game.

Finally, the application's design is such that it can easily be adapted and extended in future work. These elements create a unique putting tool combining ball tracking and elements of gamification, which has no known equivalent within the literature. The tool's value lies in its ability to enable the user to practice specific areas of their putting technique and receive instant feedback on the results.

7 Conclusion and Future Work

Over the duration of this project it was identified that the best computer vision method to track golf balls whilst putting was background subtraction. Background subtraction was tested against other techniques in a variety of different conditions, for most of the conditions background subtraction was identified to be the most successful technique when averaging the results of 102 tests. Therefore, for this putting tool background subtraction was applied for ball tracking. Another finding identified through the results of this project was that the slope of the green affects the perspective transform. In order to collect accurate distance between the ball and hole the slope of the green needs to be taken into account. Furthermore, to the gamification element of this project was targeted by comparing the users scores against previous weeks and awarding stars for each week which showed improvement as well as creating a high score tool showing the highest number of putts holed for a week. Therefore, the user was able to check whether putting progress had been made. These findings have been achieved through the following project objectives:

- Identify and implement a suitable technique for tracking the golf ball (section 4.2)
- Develop software which enables the application to measure the distance between the golf ball and the hole (section 4.2)
- The rate of deceleration of the ball can be accurately computed (section 4.3)
- The slope around the hole can be determined and visualised (section 4.3)
- Create gamification through a graphical display of the data (section 5.1)

The aim of this project, to create the foundation code for a golf-putting trainer, tracking the ball to identify the users progress over a period of time was thereby accomplished.

7.1 Contribution to the field

An important aspect of this research project includes the contribution to the relevant fields, the golf industry as well as the gamification industry. The putting training tool mainly adds value to the putting field in golf by introducing a daily tool that uses gamification for user motivation.

7.1.1 Contribution in comparison to previous putting tools

As discussed in section 1.1, tools that focus on improving putting are limited and the most frequently used putt analysis tools are "SAM putt lab", "Quintic Ball Roll" and "TOMI". These tools aim to reflect the putt movement rather than the result of the ball. Therefore, the current putting tools produce a whole different set of data than the tool created for this project. Additionally, the mentioned existing tools are used to show the user an overview of the current putting swing, and are therefore not used on a regular basis. “SAM putt lab”, “Quintic Ball Roll” and “TOMI” compare a number current putts to each other to analyse the putting swing in that particular moment in time. Meaning these tools are not seen as putting training tools, which are applied on a daily basis, but rather tools, which are used annually. As there are no previous results stored the user, cannot use these tools to track areas of improvement and see direct progress, highlighting how different “SAM putt lab”, “Quintic Ball Roll” and “TOMI” are from this tool. Hence currently there is no tool in the golf putting industry, which can be used to record the progress made over a period of time. Furthermore, the existing putting tools all lack the idea of gamification and therefore do not motivate the user through game-like elements.

7.1.2 Contribution to putting gamification

Gamification has been widely used throughout the sport industry. Although many different sports brands have created gamified sports learning tools the key elements used remain the same, motivating the user by creating an internal competition either against their own records or in comparison with their friends. These tools aim to increase participation within the targeted sport through the use of gamification. This tool adapted these gamification elements by integrating user incentives and visual comparison of the user’s progress over a period of time. Additionally, this area will be further discussed in future work. Overall, in the golf putting industry no learning tool has been introduced which integrates gamified elements therefore this tool could be a contribution of the gamified putting tools.

7.2 Future work

In order for this tool to maintain relevant within the golf putting industry it must hold the ability to be further developed and improved. Further areas of research, which would increase the effectiveness, are highlighted below.

7.2.1 Code specific calculations

In order to create a fully functioning application further work to continuously improve the app would need to be considered. The perspective transform calculations should be archived within the code, meaning that instead having to adjust the code every time a new perspective is introduced, the user only needs to upload the video holding the reference square. Then the code would extract the four corners of the square through a shape identification function. Those four points would be used to transform the reference square back into a perfect square. As the reference square is of known size the code could then calculate the height and width of each pixel in cm. This would be important because the user would only need to input the video and all calculations would be computed on its own, turning this tool into a fully functioning application.

7.2.2 Application to different areas

As mentioned in section 6.3 this tool could be adapted to suit other areas of golf, such as chipping. The same concept would be applied identifying the distance the ball is from the hole. As chipping is a relatively short shot the ball rolls at the end. Meaning there would not need to be major adjustments to the tool. However, integrating it into many different areas of golf would allow the user to identify not only putting progress but also chipping progress. Furthermore, training tools for chipping shots have not been developed yet. Only swing analysis tools have been developed, meaning it would be a further contribution to the field of golf. If the tool could be adapted into other areas of golf it could also be adapted within different sports.

Boccia is a game where multiple players aim to throw two balls as close as possible to a placed reference ball. As this application can calculate the distance between two objects it can be used as a tool to identify which player won the round. This would be beneficial for the game as it would save time as well as reduce referee bias.

7.2.3 Further gamification

Adding more interactive elements such as specific exercises with conditions, which the user needs to complete, could increase the gamification element. These exercises could then help identify the overall putting level of the user, as each exercise would test a different element of putting. Exercises would include:

- Short putts (1meter and under)
- Medium putts (2-5 meter)
- Distance control putts (over 5 meter)
- Performing under pressure (hole 20 50cm putts in a row, increase the number of putts which need to be holed and the putt distance)

Each of these exercises would define the number of putts to be holed and the distance the ball is away from the hole. The user could then track the putting progress over a period of time for each of these exercises. To further enhance the gamification element scores of exercise could be compared to the scores of friends in order to foster competition to increase motivation.

References

[1] H. Abachi and G. Muhammad, "The impact of m-learning technology on students and educators", Computers in Human Behavior, vol. 30, pp. 491-496, 2014.

[2] C. Helf and H. Hlavacs, "Apps for life change: Critical review and solution directions", Entertainment Computing, vol. 14, pp. 17-22, 2016.

[3] W. Leow, R. Wang and H. Leong, "3-D–2-D spatiotemporal registration for sports motion analysis", Machine Vision and Applications, vol. 23, no. 6, pp. 1177-1194, 2011.

[4] W. van Lier, J. van der Kamp and G. Savelsbergh, "Perception and Action in Golf Putting: Skill Differences Reflect Calibration", Journal of Sport and Exercise Psychology, vol. 33, no. 3, pp. 349-369, 2011.

[5] K. Palmer, S. Chiviacowsky and G. Wulf, "Enhanced expectancies facilitate golf putting", Psychology of Sport and Exercise, vol. 22, pp. 229-232, 2016.

[6] C. Marquardt, "The SAM PuttLab: Concept and PGA Tour Data", International Journal of Sports Science and Coaching, vol. 2, no. 1, pp. 101-120, 2007.

[7] "Putting Biomechanics", Quinticballroll.com, 2017. [Online]. Available: `http://www.quinticballroll.com/Quintic_Ball_Roll_Examples.html`. [Accessed: 10- Apr- 2017].

[8] S. Mackenzie and D. Evans, "Validity and reliability of a new method for measuring putting stroke kinematics using the TOMI® system", Journal of Sports Sciences, vol. 28, no. 8, pp. 891-899, 2010.

[9] J. Ren, J. Orwell, G. Jones and M. Xu, "A general framework for 3D soccer ball estimation and tracking", 2004 International Conference, vol. 3, pp. 1935-1938, 2004.

[10] M. Shah, Fundamentals of computer vision, 1st ed. University of Central Florida, 1997.

[11] D. King, F. Greaves, C. Exeter and A. Darzi, "'Gamification': Influencing health behaviours with games", JRSM, vol. 106, no. 3, pp. 76-78, 2013.

[12] S. Deterding, D. Dixon, R. Khaled and L. Nacke, "From game design elements to gamefulness: defining gamification", design elements to gamefulness: defining gamification. InProceedings of the 15th international academic MindTrek conference: Envisioning future media environments, pp. . 9-15, 2011.

[13] H. Lee and J. Cho, "What Motivates Users to Continue Using Diet and Fitness Apps? Application of the Uses and Gratifications Approach", Health Communication, pp. 1-9, 2016.

[14] P. Barratt, "Healthy competition: A qualitative study investigating persuasive technologies and the gamification of cycling", Health and Place, pp. 1-9, 2016.

[15] H. Park and J. Bae, "Analysis and survey of Gamification", Advanced Science and Technology Letters, vol. 39, no. 2013, pp. 2287-1233, 2013.

[16] B. Zatwarnicka-Madura, "Gamification as a Tool for Influencing Customers' Behaviour", World Academy of Science, Engineering and Technology, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering, vol. 9, no. 5, pp. 1461-1464, 2015.

[17] D. Forsyth and J. Ponce, Computer vision, 2nd ed. Boston: Pearson Education, 2012.

[18] M. Nixon and A. Aguado, Feature extraction and image processing for computer vision, 1st ed. Amsterdam: Elsevier, Academic Press, 2013.

[19] S. Challa, M. Morelande, D. Mušicki and R. Evans, Fundamentals of object tracking, 1st ed. Cambridge: Cambridge University Press, 2011.

[20] R. Radke, Computer vision for visual effects, 1st ed. Cambridge: Cambridge university press, 2013.

[21] C. Brown and D. Billard, Computer vision, 1st ed. New Jersey: Prentice-Hall, 1982.

[22] M. Sonka, R. Boyle and V. Hlavac, Image processing, analysis and machine vision, 4th ed. Stamford: Kluwer Academic Publishers, 2015.

[23] D. Senthamaraikannan, S. Shriram and J. William, "Real time color recognition", International journal of innovative research in electrical, electronics, instrumentation and control engineering, vol. 2, no. 3, pp. 1251-1253, 2014.

[24] E. Davies, Machine Vision: Theory, Algorithms, Practicalities, 4th ed. Amsterdam : Elsevier, 2012, pp. 320-325.

[25] D. Patel and S. Upadhyay, "Optical Flow Measurement using Lucas Kanade Method", International Journal of Computer Applications, vol. 61, no. 10, pp. 6-10, 2013.

[26] R. Jain, R. Kasturi and B. Schunck, Machine vision. New York [u.a.]: McGraw-Hill, 1995, pp. 315-316.

[27] Y. Seo, S. Choi, H. Kim and K. Hong, "Where are the ball and players? Soccer game analysis with color-based tracking and image mosaick", In International Conference on Image Analysis and Processing, vol. 120, pp. 196-203, 1997.

[28] K. Nummiaro, E. Koller-Meier and L. Van Gool, "An adaptive color-based particle filter", Image and Vision Computing, vol. 21, no. 1, pp. 99-110, 2003.

[29] D. Pandya and M. Zaveri, "A novel framework for semantic analysis of an illumination-variant soccer video", EURASIP Journal on Image and Video Processing, vol. 1, pp. 1-12, 2014.

[30] G. Pingali, Y. Jean and I. Carlbom, "Real time tracking for enhanced tennis broadcasts", Computer Vision and Pattern Recognition, vol. 199823, pp. 260-265, 1998.

[31] Y. Ohno, J. Miura and Y. Shirai, "Tracking players and estimation of the 3D position of a ball in soccer games", Pattern Recognition 15th International Conference, vol. 1, pp. 145-148, 2000.

[32]] X. Tong, H. Lu and Q. Liu, "An effective and fast soccer ball detection and tracking method", Pattern Recognition 17th International Conference, vol. 4, pp. 795-798, 2004.

[33] X. Tong, T. Wang, W. Li and Y. Zhang, "A novel algorithm for effective ball tracking", International Journal of Pattern Recognition and Artificial Intelligence, vol. 24, no. 03, pp. 359-379, 2010.

[34] J. Mao, D. Mould and S. Subramanian, "Background subtraction for realtime tracking of a tennis ball", VISAPP, vol. 2, pp. 427-434, 2007.

[35] M. Archana and M. Geetha, "Object Detection and Tracking Based on Trajectory in Broadcast Tennis Video", Procedia Computer Science, vol. 58, pp. 225-232, 2015.

[36] G. Gomez, P. Herrera López, D. Link and B. Eskofier, "Tracking of Ball and Players in Beach Volleyball Videos", PLoS ONE, vol. 9, no. 11, p. e111730, 2014.

[37] A. Djekoune, K. Messaoudi and K. Amara, "Incremental circle hough transform: An improved method for circle detection", Optik - International Journal for Light and Electron Optics, vol. 133, pp. 17-31, 2017.

[38] D. Ioannou, W. Huda and A. Laine, "Circle recognition through a 2D Hough Transform and radius histogramming", Image and Vision Computing, vol. 17, no. 1, pp. 15-26, 1999.

[39] M. Rizon, Y. Haniza, S. Puteh, A. Yeon, M. Shakaff, S. Abdul Rahman, M. Sugisaka, Y. Sazali, M. Rozailan and M. Karthigayan, "Object detection using

circular Hough transform.", American Journal of Applied Sciences, vol. 2, no. 12, pp. 1606-1609, 2005.

[40] T. D'Orazio, N. Ancona, G. Cicirelli and M. Nitti, "A ball detection algorithm for real soccer image sequences", 16th International Conference, vol. 1, pp. 210-213, 2002.

[41] T. D'Orazio, C. Guaragnella, M. Leo and A. Distante, "A new algorithm for ball recognition using circle Hough transform and neural classifier", Pattern Recognition, vol. 37, no. 3, pp. 393-408, 2004.

[42] T. Atherton and D. Kerbyson, "Size invariant circle detection", Image and Vision Computing, vol. 17, no. 11, pp. 795-803, 1999.

[43] X. Yu, H. Leong, C. Xu and Q. Tian, "Trajectory-Based Ball Detection and Tracking in Broadcast Soccer Video", IEEE Transactions on Multimedia, vol. 8, no. 6, pp. 1164-1178, 2006.

[44] F. Mai, Y. Hung, H. Zhong and W. Sze, "A hierarchical approach for fast and robust ellipse extraction", Pattern Recognition, vol. 41, no. 8, pp. 2512-2524, 2008.

[45] X. Wang, V. Ablavsky, H. Shitrit and P. Fua, "Take your eyes off the ball: Improving ball-tracking by focusing on team play", Computer Vision and Image Understanding, vol. 119, pp. 102-115, 2014.

[46] K. Aires, A. Santana and A. Medeiros, "Optical flow using color information: preliminary results", ACM symposium on Applied computing, pp. 1607-1611, 2008.

[47] A. Chauhan and P. Krishan, "Moving object tracking using gaussian mixture model and optical flow", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 4, pp. 1243-1246, 2013.

[48] F. Poiesi, F. Daniyal and A. Cavallaro, "Detector-less ball localization using context and motion flow analysis", International Conference on Image Processing, pp. 3913-3916, 2010.

[49] R. Luo, R. Mullen and D. Wessell, "An adaptive robotic tracking system using optical flow", Robotics and Automation, vol. 1988, pp. 568-573, 1988.

[50] E. Castillo, Z. Beltrán and J. Zannatha, "Soccer Ball Speed Estimation Using Optical Flow for Humanoid Soccer Player", Electronics, Robotics and Automotive Mechanics Conference, pp. , 178-183, 2011.

8 Appendices

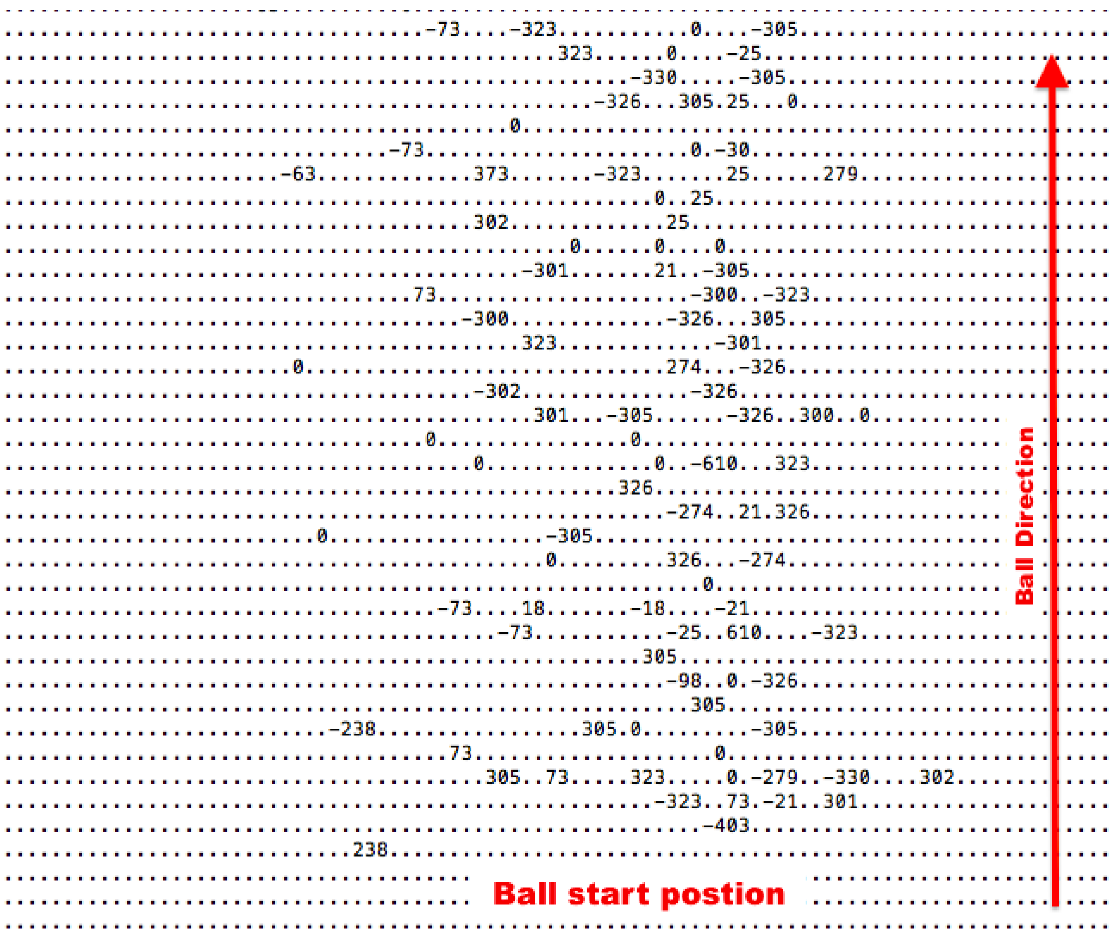
8.1 Appendix A:Video 32 - Distance from the ground truth

Video 32	Distance from Ground Truth (cm)				
Frame	Ellipse	Background Subtraction	Colour Recognition	Optical Flow	Hough Circle Transform
1	3.90	2.93	3.90	4.88	4.39
2	2.44	2.44	2.93	5.37	5.37
3	2.44	1.46	2.44	6.34	6.83
4	0.00	1.46	1.95	5.86	7.81
5	1.46	2.93	4.39	7.32	11.40
6	2.93	0.73	3.42	10.99	5.37
7	0.73	1.46	3.90	5.37	6.83
8	0.49	3.90	1.95	5.86	2.93
9	0.00	2.44	5.37	4.88	7.81
10	0.00	3.42	2.93	6.83	2.44
11	0.98	1.46	4.39	4.39	4.88
12	3.90	0.00	2.44	6.34	7.32
13	1.95	0.49	3.42	4.88	5.37
14	0.49	2.44	4.88	5.86	40.72
15	0.73	3.90	3.90	4.39	43.74
16	1.46	0.49	2.44	7.81	23.60
17	0.98	0.49	2.44	7.32	5.37
18	0.73	2.44	3.42	4.39	7.81
19	1.46	1.46	4.88	4.88	32.30
20	0.73	1.95	3.42	6.83	8.30
21	3.90	0.98	4.88	3.90	10.45
22	0.00	2.44	2.93	4.39	8.30
23	0.73	0.98	3.90	6.34	20.64
24	0.00	3.90	5.37	4.39	7.81
25	0.49	1.46	4.39	5.86	4.88
26	0.00	1.95	2.44	4.88	5.37
27	1.46	0.98	3.42	2.44	7.32
28	0.98	0.00	3.90	6.34	7.32
29	3.90	0.98	5.37	5.86	8.30
30	1.46	2.44	4.88	10.45	3.42
31	1.95	0.98	3.42	6.83	3.90
32	1.46	0.00	2.44	5.37	7.32
33	0.49	0.98	3.90	3.42	4.88
34	1.95	2.44	5.37	4.88	8.30
35	0.49	0.98	3.90	3.42	2.93
36	0.73	0.49	1.95	2.44	4.88
37	3.90	1.95	2.44	2.93	3.90
38	0.49	0.00	2.93	4.88	1.46
39	0.49	0.49	4.88	3.42	3.42
40	1.95	0.98	2.93	2.93	4.88
41	0.73	1.95	1.95	1.95	2.44
42	0.73	1.95	4.88	4.88	2.93
43	0.49	0.49	1.95	2.44	3.42
44	0.00	0.49	1.95	2.93	2.44
45	0.00	0.00	1.46	1.95	1.46

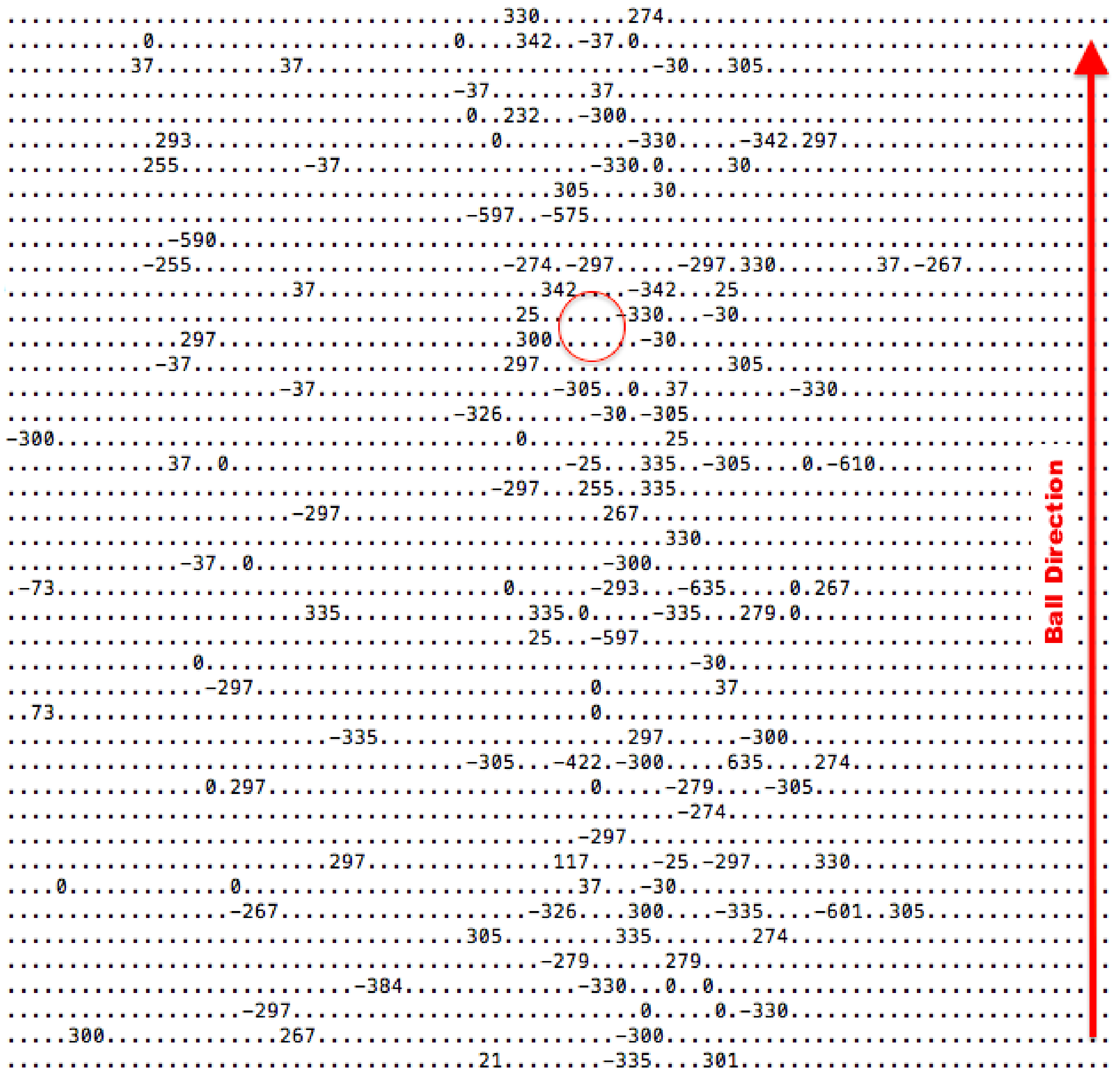
8.2 Appendix B: Ball's distance from the hole - comparison between background subtraction and the actual measurement

Video	E&BS distance from center (cm)	Measure center hole (cm)	Difference (cm)
1	0.0	0.0	0.0
2	17.5	18.5	1.0
3	20	18.5	1.5
4	0.0	0.0	0.0
5	0.0	0.0	0.0
6	40.0	38.5	1.5
7	20.5	19.5	1.0
8	0.0	0.0	0.0
9	0.0	0.0	0.0
10	39.2	37.5	1.7
11	43.6	41.5	2.1
12	25.0	23.4	1.6
13	0.0	0.0	0.0
14	60.4	56.5	3.9
15	0.0	0.0	0.0
16	0.0	0.0	0.0
17	18.4	16.0	2.4
18	0.0	0.0	0.0
19	0.0	0.0	0.0
20	0.0	0.0	0.0
21	0.0	0.0	0.0
22	0.0	0.0	0.0
23	0.0	0.0	0.0
24	0.0	0.0	0.0
25	19.2	20.5	1.3
26	24.0	22.0	2.0
27	0.0	0.0	0.0
28	27.3	25.0	2.3
29	49.8	47.0	2.8
30	0.0	0.0	0.0
31	18.6	20.5	1.9
32	0.0	0.0	0.0
33	9.6	8.5	1.1
34	45.3	43.0	2.3
35	35.0	33.0	2.0
36	45.5	43.5	20.
37	54.6	52.0	2.6
38	0.0	0.0	0.0
39	0.0	0.0	0.0
40	0.0	0.0	0.0
41	0.0	0.0	0.0
42	0.0	0.0	0.0
43	27.7	26.0	1.7
44	37.2	34.3	2.9
45	0.0	0.0	0.0
46	9.3	10.0	0.7
47	0.0	0.0	0.0
48	15.5	14.0	1.5
49	0.0	0.0	0.0
50	20.7	18.5	2.2

8.3 Appendix C: Deceleration of the golf ball in the first segment of the 1m downhill putt



8.4 Appendix D: Deceleration of the golf ball in the second (middle) segment of the 1m downhill putt



8.5 Appendix E: Deceleration of the golf ball in the third and final segment of the 1m downhill putt